

Milliyet
YAYINLARI

BİLİM DÜNYASI

ANSIKLOPEDİSİ

BİLGİSAYAR



BİLİM DÜNYASI ANSİKLOPEDİSİ

Cilt: 1
BİLGİSAYAR

Genel Müdür:
MUHİTTİN KÂZIMOĞLU

Yönetmen:
MEHMET İLKORUR

Yönetmen Yardımcısı:
HİKMET ALTINKAYNAK

Dilimize Çevirenler:
ŞULE SAKIZLIOĞLU
YAKUT GÜNERİ
ERAY CANBERK
GÜNDÜZ TUNA

Grafiker:
AYSEL UĞUR

İşletme Müdürü:
ALİ NAZIM ONARAN

Teknik Danışman:
YETKİN ALDINÇ

Ofset Hazırlama ve Baskı:
Milliyet Yayın A.Ş.
Ayazağa Tesisleri

Copyright (Yayın Hakkı):
Kesim Telif Hakları Ajansı (83)

Türkiye'de Yayın Hakkı:
© MİLLİYET YAYIN A.Ş.





BAŞLARKEN

Çağımız, bilimlerin de baş döndürücü bir hızla geliştiği bir yapıya sahip. Bu hızı ayak uydurabilmek, aydınlar için bile gittikçe zorlaşıyor.

Sunduğumuz "Milliyet Bilim Dünyası Ansiklopedisi" bu değişikliklerden en önce gelen bilim dallarını içeriyor. Her biri bir ciltte oluşturulacak olan; 1. Bilgisayar, 2. Uzay, 3. Enerji herkesin kesinlikle başvurması gereken konular. İlk cildimiz BİLGİSAYAR adını taşıyor. Yirminci yüzyıla damgasını basan bilgisayarla, bilim dünyasının bilinmeyen yönlerini tanımak, artık zor değil. Ayrıca sanayiden tarıma, eğitimden ticarete kadar her konuya hızla giren bilgisayarların, yararlanmamız gereken gerçek yönlerini öğrenmemiz "Bilim Dünyası Ansiklopedisi" ile olanaklı. Çağa, toplumumuza olan sorumluluğumuzu; bilimlerdeki olağanüstü değişiklikleri, bir ansiklopedi düzeyiyle sizlere sunmaktan gurur duyuyoruz.

1	2	3	4	5	6	7	8	9	10	LATİN
1	٢	٣	٤	٥	٦	٧	٨	٩	١٠	ARAP
∨	∨∨	∨∨∨	∨∨∨	∨∨∨	∨∨∨	∨∨∨	∨∨∨	∨∨∨	∨∨∨	BABİL
1	0	11	12	13	14	15	16	17	18	MISIR
.	..	...		—	.	..	...		==	MAYA
I	II	III	IIII	V	VI	VII	VIII	IX	X	ROMEN

Sayılar ve Sayı Düzenleri

Birden ona kadar olan sayıların altı değişik şekilde yazımı. Latin ve Arap rakamları, aynı sayı sisteminin değişik şekilleridir. İkisinde de sıfır sayısı vardır. İkisi de halen kullanılmaktadır. (Üstte)

NASIL ki alfabe bir 'harf düzeni' ise, sayıların da düzenleri vardır. Bunlar, sayıları belirtmek için kullanılan kümeler halindeki işaretlerdir. Farklı uygarlıklar, farklı alfabelerinin yanı sıra, farklı sayı düzenlerini de kullanmışlardır.

Eski Yunanlılar, sayıları ve harfleri aynı işaretlerle gösteriyorlardı. Alfabenin ilk harfi olan alfa (α) aynı zamanda bir (1) anlamına, üçüncü harf olan gama (γ) da üç (3) anlamına gelmekteydi.

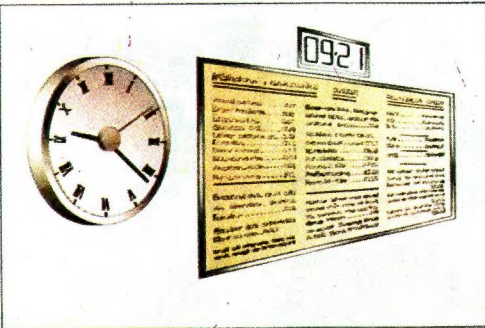
Roma Rakamları: Romalılar tarafından geliştirilen sayı düzeni günümüzde de zaman zaman kullanılmaktadır. Yunanlılar gibi, Romalılar da sayıları alfabelerinden seçmişlerdir. Örneğin V harfi 5 sayısını, X harfi 10 sayısını, L harfi 50 sayısını, C harfi 100 sayısını, ve M harfi 1000 sayısını gösteriyordu.

Diğer sayılar, bir harf dizini olarak, en büyüğünden başlamak üzere birbirine toplanacak şekilde yazılırdı.

Kendisinden büyük bir harften önce gelen bir harfse, büyüğünden çıkarılır, sayının değeri bu şekilde bulunurdu. Örneğin MM 2000, MCM 1900, MCMLXX 1980 ve MCMLXXIX 1979 sayılarının yerine kullanılırdı.

Sıfırın önemi: Sıfır, ya da hiçbir, matematikte oldukça önemli bir buluştur. Yunan ve Roma sayı düzenleriyle günümüz arasındaki en büyük fark, bugün sıfırın sayı olarak kullanılmasıdır.

Sayı düzenimizde bir rakamın değerini, bulunduğu yer belirlemektedir. Yani 50, 5'in 10 katıdır; 500, 100 katı ve 5000, 1000 katıdır. Böylece sıfır sayısı bize beşin 1 ya da 10 adet olduğunu gösterir.



Kör bir insanın kâğıt üzerindeki kabartmaları hissederek okuması. Braille sistemi diye adlandırılan bu sistem, ikili sisteme benzer: Bir noktada kabartma vardır ya da yoktur. İkili sistemde de, belirtilen noktada ya bir sayı vardır (1' ile gösterilir), ya da yoktur (0 ile gösterilir). (Sağda)



Sayı Tabanları: Günümüzde sayılar, 10'luk rakam, ya da 'basamak' ile gösterilmektedir. Nesneleri onlar, yüzler, ya da binler (V.S.) ile saymaktayız, 10'un çarpımlarıyla yapılan bu sayım **10'luk düzen**, ya da '**ondalık sayı**' adı ile adlandırılır.

10'luk düzende, 10 işareti on sayısını gösterir. (1=10 sayısını, 0=hiçbir ekstra basamak olmadığını gösterir.) Eğer 8'lik düzeni kullanmış olsaydık, 10 işareti sekiz sayısını gösterecekti. (1=8, sayısını). 8'lik düzende on sayısını 12 işareti ile gösteririz. (1=8 sayısını, 2=2 ekstra değer.) 8'lik düzen, 'oktal düzen' olarak adlandırılır ve bu düzende sayıları oluşturmak için sadece sekiz işaret kullanılır.

Bilgisayar, iki basamaklı olan ikilik düzenle çalışır. Yani ikilik düzende sadece iki işaret, 0 ve 1 vardır. Sayma işlemi bu iki işaretle gerçekleştirilir. 138 sayfalık bir kitabın sayfa sayısı ikilik düzenle gösterilirse, 110000 dir. Çok büyük olmayan sayılar bile, ikilik sistemde birçok basamakla yazılır. Örneğin bin sayısı 1111101000 ile gösterilir. Bu durumdan da anlaşıldığı gibi, ikilik düzen, günlük yaşantımızda kullanılacak kadar pratik değildir.

Bizim sayı ve para sistemimiz 'onlu', ya da 'ondalık' sistemdir. Ağırlık ve uzunluk ölçümlerinde, metrik sistem de, onlu sistem'i kullanır; ama daha eski ölçümler (arşın, endaze gibi) başka sayı sistemlerini kullanılır. Metrik sistem, hesaplama işlemlerinde, özellikle burada görüldüğü gibi otomatik ağırlık ölçme ve etiketleme işlerinde büyük kolaylık sağlar. (Ortada)



Eski Mezopotamyalılar, sayı sistemlerini, 'altmışlık sistem' e göre düzenlemişlerdi. (Altta)

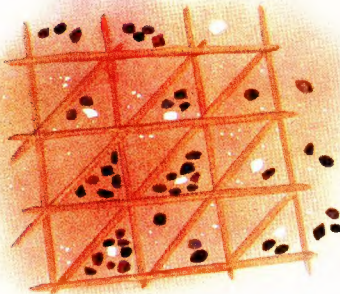


Dijital bilgisayarlar, bir elektrik uyarısının varlığı ya da yokluğuna göre çalışır. İkili düzende, 0 ve 1 işaretleri bilgisayar devrelerindeki elektrik akımının var olup olmadığını gösterir. İnsanlar bugün, sayım işlemlerini birçok cihaz yardımıyla yapabilmektedir. Bunların bir kısmı günlük hayatımıza girecek kadar basit, bazıları ise ancak özel işlerde kullanılacak kadar karmaşıktır.

Hesap Makineleri

DÜNYANIN her yerinde insanlar, toplama işlemlerini parmaklarını kullanarak yapmaktadırlar. Parmaklar, insanların, matematiksel işlemleri uygulayabilmek için kullanmayı öğrendikleri ilk araçtır. Parmaklar, basit işlemlerde oldukça yararlı ve pratiktirler. **Hesap Makinasından da Hızlı:** İlk ve en eski hesap aletleri (abaküs) saygınlardır. Binlerce yıl boyunca kullanılmışlardır. İlk saygılar, kum üzerine çizilen çizgilerden ibaretti. İnsanlar, sayıları bu çizgilerin üzerine koydukları küçük taşlarla gösteriyorlar, işlemleri ise bu taşların yerlerini değiştirerek yapıyorlardı.

Bugün kullanılan en yaygın saygı ise, bir çerçeve içine yerleştirilen tahta ya da metal çubuk-



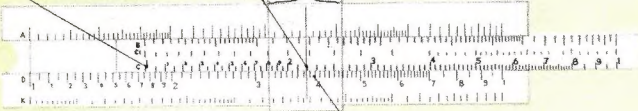
Abaküs iyi bir kullanıcının elinde, hâlâ bir elektronik hesap makinasından çabuk işlem yapar. (Ortada)

Birkaç taş ve çizgiyle, çok basit bir hesap makinası yapabiliriz. Yanda görülen 341x123 işlemini yapıyor. Üstte üç artı dört artı bir taş, 341'i gösteriyor. Yukarı dan aşağıya bir artı iki artı üç taş 123'ü gösteriyor. Her kutudaki taşlar, hemen yanındaki ve üstündeki taşların çarpımını belirtiyor. Yani sağ üst köşeye bir (1x1) taş koyuyoruz. Üstte ortadaki kutuya ise dört (4x1) taş koyuyoruz, vs. sonuç, her çaprazdaki kutulardaki sayıların toplanmasıyla elde ediliyor. Burada sonuç 41943 çıkmış. (Altta)

1. Önce cetvel ayarların. C tarafındaki 1, D tarafındaki 1.76'ın üzerine getirilir.

2. Gösterge yerine getirilir. Ortadaki çizgi, C tarafındaki 2.18'in üzerine getirilir.

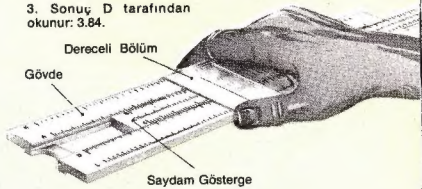
HESAP CETVELİNİN KULLANIMI



1.76x2.18 hesabını yapmak için hesap cetvelinin kullanılması. Sonuç, 3.84 çıkarıyor. Çok pratik hesaplarda, bu kesin sonuç olan 3.8368'e çok yakındır. 176x0,218 = 38,4 hesabı da aynı şekilde yapılır. Virgölün nerede olacağını bulmak için, kabaca bir hesap yapmak gerekir. (Üstte)

Hesap cetvelinin üç temel parçası vardır: Gövde, üzerinde sayılar bulunan hareket eden parça, ve tam ortasında ince bir çizgi bulunan şeffaf gösterge (Sağ'a,

3. Sonuç D tarafından okunur: 3.84.



Gövde

Dereceli Bölüm

Saydam Gösterge

lardan oluşmaktadır. Her çubuk üzerinde, belli sayıda hareket edebilir boncuklar bulunur. Sayılar, boncukların belli yerlere getirilmesiyle gösterilir. Bazı sayaçlar, iki ayrı parçadan oluşmaktadır. Bu tip sayaçlarda üst kısımdaki boncuklardan her biri beş birim sayıyı temsil eder.

Sayaç kullanmada usta olan bir kişi, en uzun ve karmaşık işlemleri bile bir hesap makinasından hızlı yapabilir. Sayaçlar, bugün de dünyanın birçok yerinde, marketlerde, işyerlerinde ve hatta bankalarda yaygın bir şekilde kullanılmaktadır.

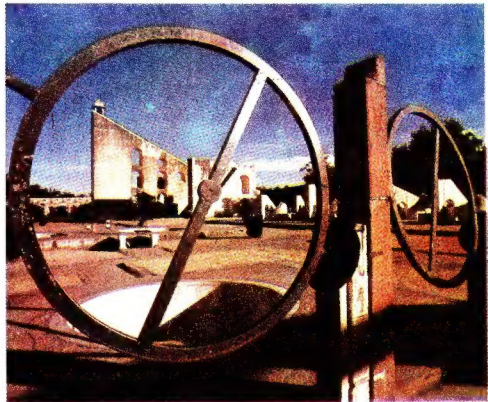
Hazır Cetveller: Matematikte kullanılan araçların en kullanışlılarından biri de, hazır cetvelerdir. Bunlar bir işlemin kısa yorumunu gösterir. İşlemi biz yapacağımıza, doğrudan hazır cetveldeki sonucu bakabiliriz.

Her işlem için değişik hazır

Gözlem aletleri, eskiden bir hesap aracı olarak kullanılıyordu. Yanda görülen, Hindistanda gökyüzündeki yıldızların yerini hesaplamak için kullanılırdı. (Alitta)

cetveller hazırlanır. Örneğin Santigrat dereceleri, Fahrenheit dereceleri çeviren, açılarını sipüs ve kosinüslerini gösteren cetveller vardır. İngiltere, 1971 yılında metrik sisteme geçerken, halk eski ve yeni sistem arasındaki oranları gösteren cetvelleri sürekli yanında taşırdı.

Toplayarak Nasıl Çarpılır: Logaritma, çok uzun sayıları çarparken, ya da bölerken kullanılan bir çeşit hesap aracıdır. Bu sayıların sadece logaritmasını toplayarak, ya da çıkartarak, onları çarp ya da böleriz. Herhangi bir sayının logaritmasını bulmak için hazırlanmış cetveller vardır. İki





Polis, bilgisayara çalınan arabaların listesini gönderir.



Ehliyet ofisi, her arabanın ve sürücünün hakkında bilgi gönderir.



Dedektifler, suçların ve şüpheliler hakkında bilgi yollar.



Güvenlikteki bilgisayar, arabalar ve sürücüler hakkındaki bilgileri saklar.



Polisteki bilgisayar suçların ve şüpheliler hakkındaki bilgileri saklar.



Polis, sürücüler hakkındaki bilgileri bilgisayardan alır.

Ofiste çalışanlar, sürücülerin şüpheli olup olmadığını inceler.

Polis merkezi, arabanın çalınıp çalınmadığını inceler.



Pölis, ehliyetleri inceler ve ayrıntıları merkezden sorar.



Polis arabası. Şüpheli insanlar görünce, merkezle ilişki kurar



Telefondan arayan biri, günlerce yerinden alınmayan bir arabayı söyler

sayıyı çarpmak için, önce cetvel- den iki sayının da logaritmasını buluruz. İki logaritmanın toplamı, çarpımın logaritmasını verir. Asıl çarpımı bulmak için de bu kez cetveli tersine kullanmamız gerekir.

Bölme işlemi de, buna benzer bir şekilde yapılır. İki sayıyı birbirine bölmek yerine, o sayıların logaritmalarını birbirinden çıkarır, bölümün logaritmasını elde ederiz.

Hesap cetveli de, hemen hemen aynı şekilde çalışır. Özellikle çabuk ve yaklaşık hesaplamalarda kullanılır. Hesap makineleri, büyük sayılarla yapılan işlemlerde daha kesin sonuçlar

verdiği halde, birçok mühendis, bilim adamı ve grafiker, kullanıma daha pratik olan hesap cetvelini yeğler.

Bilgisayara Giriş

BİLGİSAYARLAR, değişik boyutlarda çeşitli işlerde kullanılmak üzere yaşamımıza girmiştir. Basit hesaplar yapan ucuz ve küçük makinelerden; herhangi bir insanın yaşamı boyunca yapamayacağı kadar büyük problemleri bir dakikada çözebilen, karmaşık, milyonluk makinalara kadar çeşitleri vardır.

Bilgisayarların çok iyi yaptığı

iki farklı iş vardır. Birincisi, işlemleri çok çabuk ve kesin yapabilirler. İkinci de, epey büyük miktarlarda dökümanları saklayabilir ve istenildiğinde çok kısa bir zamanda sunabilirler.

Büyük Makinalar: Elektronik bilgisayarların gelişmesi, İkinci Dünya Savaşı sıralarına rastlar. Amerikan ordusu, bombaların ve mayınların yönlerini saptayabilmek için bir araç ısmarlar. Adı ENIAC (Elektronik Sayısal Entegrasyon ve Hesap Makinası) olan bu araç, çok büyük bir makineydi ve yapımı 1946 yılında gerçekleştirildi.



Günümüzün modern fotoğraf makinelerinin içine küçük bilgisayarlar eklenmiştir. Saniyeden çok daha kısa bir süre içinde çevredeki ışığı ölçerek, filmin üzerine düşürülecek ışık miktarını bulurlar. Fotoğraf makinelerinin mercekleri de, yine bilgisayarların yardımıyla yaratılır(Üstte).

ENIAC'ın, 18.000 elektronik merkezi vardı ve 150 kilowattlık güçle çalışıyordu. Ama büyüklüğüne karşın çok ilkel bir makinaydı. Günümüzün küçük bir bilgisayarındaki çok daha yavaş işliyor ve hiçbir programı bünyesinde saklayamıyordu.

Günümüzün bilgisayarları, istediğimiz her şeyi yapabildikleri için, bize çok faydalıdır. Onlara sadece, program dediğimiz komut dizisi vermemiz yeterlidir. İlk programlanabilir bilgisayar, 1949 yılında gerçekleştirilmiştir.

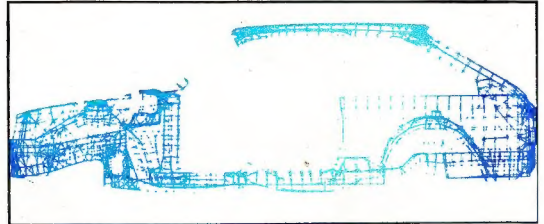
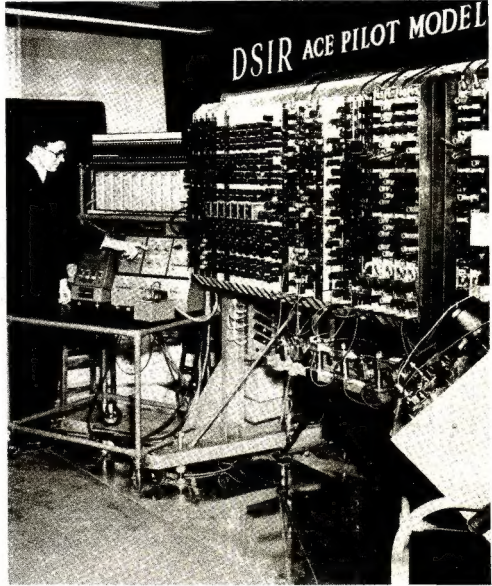
Kısa bir zaman sonra, yapımcıları, bu bilgisayarların bakımını kaldırabilecek her kuruluşa satış yapmaya giriştiler. İlk önce Amerika'da UNIVAC (Evrensel Otomatik Bilgisayar) şirketi kuruldu. Bu şirket, bilgi girişi için ilk kez manyetik teyp kullandı. Teyp üzerine sayıların yanında harfleri de yazdırabiliyordu.

Özellikle iş amaçları için hazırlanan bilgisayarları, ilk kez Lyons adında İngiliz bir firma gerçekleştirdi. Büyük bir öngörülülikle, 1947 yılında Lyons firması, Lyons Elektronik ofisini (LEO) kurdu. Bu ofis, bilgisayarlarının kullanımında, iş adamlarına yol gösteriyordu.

Ucuz ve güçlü: Bugün, dünyada yüzlerce çeşit bilgisayar kullanılıyor. İlk bilgisayarların elektronik merkezleri, yerini kısa bir süre sonra daha güvenilir ve harcamasız olan transistörlere bıraktı. Sonra transistörlerin yerini de, **çip** adı verilen ucuz entegre devreler aldı. Bu entegre devrelerin içine, çok küçük bir alana, çok fazla sayıda devre elemanı yerleştirilmiştir.

Bu teknik kolaylıklar sonucu, bugün bir cebe sığacak büyüklükte bir bilgisayarın kapasitesi, ENIAC ve UNIVAC'ın toplam kapasitesinden daha büyük olabiliyor.

Bilgisayarların bulunuşu, XVIII. yy. endüstri devriminden sonra gerçekleştirilen en önemli gelişmedir. Bugün bile, dünyada pek çok insanın hayatında önemli bir noktayı oluşturmaktadır.

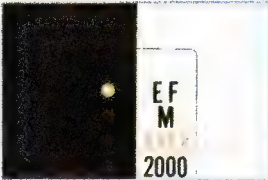
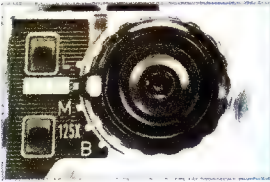


1950 yılında yapılan, "ACE" adlı bu bilgisayar, bir dizi elektronik silindirlere oluşuyordu. Daha sonraki bilgisayarlarda ise, bu silindirlerin yerini transistörler aldı (En Üstte).

Mühendisler ve tasarımcılar da piyasaya çıkarılacak yeni ürünler için bilgisayarlardan yararlanırlar. Örneğin, piyasaya çıkarılacak yeni bir otomobile ilgili bilgiler bilgisayara verildikten sonra, ekrana yeni ürünün üç boyutlu görüntüsü yansıtılır (Üstte).



Modern bir fotoğraf makinesinin, ışık ölçüm sistemi (pozmetre), "Quarz" kristallerinden oluşmaktadır (üstte).



Bazı fotoğraf makineleri, ışık ölçümünü içinde bulunan bir mini bilgisayarın yardımıyla gerçekleştirir. Bu bilgisayar, ışığın oranını, saniyenin 1/1000 süratinde ölçebilir (Küçük resimler)

ILK yapılan bilgisayarlar-
dan bazıları, işlemleri
ondalıklı düzenle yapıyor-
lardı. Ancak bu düzenin fazla
güvenilir ve kesin olmayan yavaş
ve pahalı bir yol olduğu kanıt-
landı. Ondalıklı düzen, genelde
yeterli olarak görülür. Oysa,
aslında bizler de en az onun kadar
yavaşız ve yanlış yapmaya yöneli-
miş. Bu yüzden, insan eliyle yapı-
lan işlemlerde bu yetersizlik göze
çarpacak kadar etkili olmaz.

Bilgisayar yapımı başladıktan
çok kısa bir süre sonra, elektro-
nik ve elektromekanik aletlerde
tek güvenilir işlem şeklinin açık-
kapalı devre modeli üzerinde
yapılabildiği anlaşıldı. Bu mode-
lin iki koşulu olduğuna göre,
işlemlerin iki rakamlı bir düzenle
yapılması kaçınılmazdı.

İkilik düzende, yalnızca iki
rakam vardır: '0' ve '1'. Ama bu
genelde işlemlerin doğasını de-
ğiştirmez. Onluk düzende yazıla-
bilen her sayı, ikilik düzende de
yazılabilir. Yapı olarak da birbi-
lerine çok benzerler.

Örneğin:

Onluk düzende $1984 = 1 \times 10^3$
+ 9×10^2 + 8×10^1 + 4×10^0 ise,
aynı şekilde ikilik düzende de
 $1011 = 1 \times 2^3$ + 0×2^2 + 1×2^1 + 0
diye yazabiliriz.

BOOLEAN CEBİRİ

IKİLİK düzen kullanılmaya
başlandıktan sonra, ikinci
sorun gündeme geldi. Arit-
metik işlemleri de ikilik düzene

göre yapmak gerekiyordu. Daha
önce kullanılan elektronik sayım
metotlarının çok yavaş ve pahalı
olduğu bir gerçektir. Bu sorun,
George Boole tarafından çözümlen-
di. Yaratıcısının adıyla anılan
'Boolean Cebiri', iki koşullu bir
cebir düzeni olmakla birlikte,
'ikilik düzen aritmetiği' ile doğrudan
bir bağlantısı yoktur. Ancak,
kesin kuralları olan herhangi bir
'iki koşullu düzen'de de kullanı-
labilecek kadar yetkin ve kulla-
nışlı bir yöntemdir. İkilik düzen,
günümüzde kavranma açısından
bir sorun yaratmamaktadır. Dahası
temel eğitim evresinde bile bu
düzeni öğreten pek çok
ülke vardır.

İkilik düzen aritmetiğinde
gerekli olan kesin koşulları belir-
leyebileceğimize göre, bu düze-
nde Boolean Cebirini kullanabili-
riz. Aynı zamanda bu yöntem
bize bilgisayar elektronigi ince-
lerken de oldukça yararlı olabilir.

Normal cebir, aritmetiğin bir
genellemesi olduğu için normal
aritmetiğin kurallarını izler. Boolean
Cebiri ikilik cebire çok ben-
zer, ama değişken ve sabit sayılar
ancak iki değerde olabilir: '0' ve
'1'. Boolean Cebiri, ikilik aritme-
tiğin tüm kurallarına uymaz; bu
yüzden de ikilik cebirin bir genel-
lemesi sayılmaz.

Şimdi Boolean toplamı ve
çarpımı ile ikilik toplam ve çar-
pımı karşılaştıralım:

İKİLİK TOPLAM

	0	1
0	0	1
1	1	0

Artık

İKİLİK ÇARPIM

	0	1
0	0	0
1	0	1

BOOLEAN TOPLAMI

	0	1
0	0	1
1	1	1

BOOLEAN ÇARPIMI

	0	1
0	0	0
1	0	1

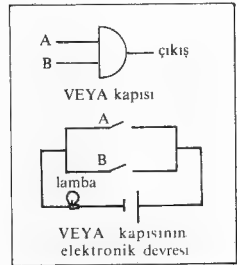
Bilgisayardaki bütün devre-
ler, bu cebir mantığı üzerine
kurulmuştur. İşlemler bu mantık
üzerinden yapılır. Şimdi biz bu
devrelere girecek değiliz. Ama
mantıkta kullanılan bazı kapıları
inceleyeceğiz ki, bu kapılar bilgi-
sayardaki elektronik devrelerin
temel işleyişini gösterir.

Bizim inceleyeceğimiz üç
kapı, **Boolean Cebiri**'nin üç temel
işlemini gösterir. Bunlar çarpma
işlemini gösteren **VE** kapısı, to-
plama işlemini gösteren **VEYA**
KAPISI ve giren elemanı değiştiren
DEĞİL KAPISIDIR. Bu kapıların
bir ya da birden fazla girişi ve tek
çıkışı vardır. İşlem içinde kullanı-
lacak değerler girişte gösterilir,
sonuç da çıkışta yazılır.

Bu kapıların birçok gösterişi
şekli vardır. Biz bunlardan bir
tanesini benimseyerek, örnekle-
meye geçelim.

VEYA KAPISI

Boolean Cebirinde toplama
işlemini gösterir. Yalnız bildi-
ğiniz gibi Boolean aritmetiğinde
toplama işlemi, normal toplama
işlemine benzemez. Örneğin
 $1+1=1$ sonucu normal aritme-
tikte doğru olmadığı halde Boolean
aritmetiğinde doğru ve tek
sonuçtur.



Şimdi de devreye bakalım.
Lambanın yanması için A ya da B
anahtarlarından birinin kapalı
olması yeterlidir. Sonuçta, biz
VEYA kapısını paralel bağlanmış
bir devreye benzetebiliriz. Akımın

geçebilmesi için, tek bir kolun kapalı olması yeterlidir.

VEYA kapısının elektronik devresine göre bir şema hazırlarsak:

A	B	LAMBA
Açık	Kapalı	Yanmaz
Açık	Kapalı	Yanar
Kapalı	Açık	Yanar
Kapalı	Kapalı	Yanar

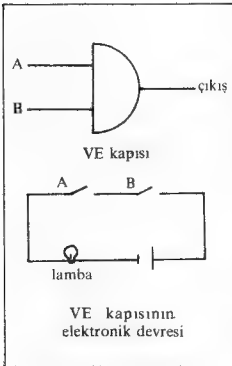
Her kapının, Boolean aritmetiğinde karşılığı olan bir denklem vardır. **VEYA** kapısının Boolean denklemi de şudur:

Fonksiyon $(f)=A+B$

Buradaki fonksiyon (f), çıkışı belirtmektedir.

VE KAPISI

VE kapısı, Boolean aritmetikinde çarpma işlemi gösterir. Yani bu kapının Boolean denklemi olarak karşılığını $f=A.B$ olarak yazabiliriz.



Şimdi yine devreyi inceleyelim. Akımın geçmesi için hem A, hem de B, anahtarlarının kapalı olması gerekir. Yani **VE** kapısını

da seri bağlanmış bir devreye benzetebiliriz. Lambanın yanması için, daha önce de belirttiğimiz gibi iki anahtar da kapalı olmalıdır. Buna göre yine bir şema hazırlayabiliriz.

A	B	LAMBA
Açık	Kapalı	Yanmaz
Açık	Kapalı	Yanmaz
Kapalı	Açık	Yanmaz
Kapalı	Açık	Yanmaz
Kapalı	Kapalı	Yanar

Bu elektronik devreleri bilgisayar mantığıyla, daha yakından bağdaştırmaya çalışırsak, şöyle diyebiliriz: Bilgisayarda iki koşul vardır. Bunlardan biri sıfır '0', diğeri de bir '1'dir. Boolean Cebiriyle elimizdeki elektronik devreleri karşılaştırsak, girişlerdeki 'açık' ve 'kapalı' konumlarına karşılık olarak '0' ve '1' konumlarını koyabiliriz. Aynı şekilde 'yanmaz' konumuna '0', 'yanar' konumuna da '1' diyebiliriz. Böylece bilgisayardaki giriş ve çıkış konumlarını da belirlemiş oluruz.

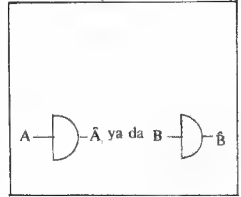
Bu kapılara ikiden fazla giriş varsa da, çıkış konumları değişmez. Yani **VEYA** kapısının girişinde bulunan herhangi bir '1', çıkışı '1' yapmaya yeterlidir. Buna karşılık **VE** kapısının girişindeki bir '0' bile, çıkışın '0' olmasına neden olur.

DEĞİL KAPISI

DAHA önceden de belirttiğimiz gibi, Boolean Cebirinin üç temel kapısı vardır. Bunlardan ikisini biraz önce inceledik. Şimdi de üçüncü kapı olan

DEĞİL kapısını inceleyelim.

Değil kapısının görevi, girişte gelen koşulu değiştirmektir. Her türlü çizim ve denklemde, değil A 'A' işaretiyle, değil B 'B' işaretiyle gösterilir. Bunu değil kapısının çizimiyle birlikte gösterirsek:



Ayrıca $1=0$, ve $0=1$ 'dir.

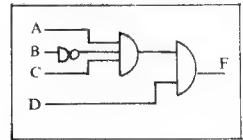
Değil kapısının tek girişi vardır. İşlevi çok basit olmakla birlikte, Boolean Cebirinde çok önemlidir. Birçok işlemi basite indirmekte bize yardımcı olur.

KAPILARIN BİRLİKTE KULLANILMASI

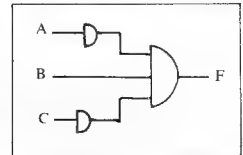
NASIL ki bir cebir denklemi toplamayı, çıkartmayı, çarpmayı ve bölmeyi bir arada yapıbiliyorsak, Boolean Cebirinde de bu üç kapı bir arada kullanılabilir. Şöyle bir denklem düşünelim:

$$f=(A+B+C)'D$$

Bu denklemin çizimi şu şekilde olacaktır.



Aynı şekilde bir çizimi de denkleme çevirebiliriz.



Denklem:
 $f=A+B+C$

ÇİP

Daha önce, bilgisayarın yaşantımızdaki önemine değinmiştik. Ayrıca bu buluşun çok yeni olduğu ve korkunç bir hızla geliştiğini de belirtmiştik. Şimdi size bunun çok güzel bir örneğini sunacağız. Bilgisayarın 1940'lı yıllarda geliştirildiğini biliyoruz. O gün tonlarca ağırlıkta yapılan bilgisayarlar, bugün ufak bir kutunun içine yerleştirilebiliyor. Çip adını verdiğimiz bu kutucukların büyüklükleri ancak milimetrelerle ölçülebiliyor. Üstelik o eski büyük bilgisayarlardan çok daha becerikli, çok daha pratik. Hata yapma olasılığı da çok az.

Sadece kırk yıl içinde gerçekleştirilen bunca şeyi düşünecek olursak, çok değil beş on yıl sonra bir bilgisayarın neleri başarabileceğini tahmin etmek, pek zor olmasa gerek.



Yaşantımızı değiştiren minicik mucize Çip, yeni doğmuş bir bebeğin tırnağı büyüklüğünde. Mumu söndürebiyen bir ağız dolusu nefes, Çipin uçmasına neden oluyor. Ve ince bir cam deniz süngerinden bile daha dayanıksız.

Çip denilen gümüş renkli minik buluş, çağımıza adını verdi. En yalın biçimiyle Çip bir elektronik devre: Silikonun içine ve üzerine konan minicik düğmeler, olağanüstü incelikte ince metal "tellerle" birleştiriliyor. Mikroskopla



Yine "çip"lerin yardımıyla gerçekleştirilen bir buluş da ağır işiten insanlara yarıyor. (Ustte)

bakınca, Çipin karmaşık yüzeyi, cadde ve sokaklarıyla bir mahalleyi andırıyor.

Daha önemlisi, yarım milimetre kalınlığındaki bir Çip, bir milyon elektronik parçadan oluşuyor. 1946'da yapılan 30 tonluk ilk bilgisayar ENIAC, bunun onda biri kadar bir parçadan oluşmuştu. Oysa şimdi bilim adamları, bir Çipten bilgisayar yapmayı başardılar. Saniyede bir milyon kadar hesap yapabilen bu minik bilgisayar, çok az enerji ile çalışıyor.

Tıpkı insan beyni gibi, Çipin uygulama alanı sonsuz. Örneğin bir mikro-bilgisayar, kalp atışlarını düzenliyor, termostatları ayarlıyor, araba motorlarını kontrol edebiliyor. Ve Çipin küçük bir alana sığdırılmış beyin gücünden, artık özel bilgisayarlar yapılıyor.

ÇİPİN YAPIMI

Çipin temelini, dünyada oksijenden sonra en çok bulunan madde, silikon oluşturuyor. Çiplerde kullanılan silikon, kuvarz kayalarından alınıp, rafine ediliyor. Silikonun özellikleri, elektrik sinyalleri büyüten ve kontrol edebilen



Yeni doğmuş bir bebğin büyüklüğünde olan, "çip"ler, dev laboratuvarlarda hazırlanıyor. (Ustte)

transistör yaratmak için, değıştirebiliyor.

İlk aşamada **silikon** eritilip, uzun kristallere dönüştürülür. Sonra, ince salam dilimlerine benzer dilimlere ayrılır. Bu dilimlerin üzerine, yüzlerce **Çip** yapılır. Her bir dilim, binlerce transistörü içerir.

İnce dilimler, bir tabaka oksit ile izole edilirler. Sonra, ışığa karşı duyarlı, **fotorezist** ismindeki yumuşak bir plastikte kaplanırlar. Bir şablonla kapatılıp, ultraviyole ışığına tutulunca, ışığa açık **fotorezist** istenilen biçimde sertleşir. Asit ve eriticiler, ışığa tutulmayan oksit ve **fotorezist** çıkarılırlar. Dördüncü aşamada, daha fazla silikon ince dilimin üzerine konarak, aynı işlemler tekrarlanır. Sonra, artı ve eksi akımları geçirecek iletken bir kimyasal madde dökülür. Birbirine bağlı "pencereler" oluşturularak, önceki aşamalar tekrar edilir. Bu pencerelerin içine aliminyum doldurularak, iletken yollar oluşturulur. Her bir **Çip**, transistörüyle birlikte ince dilimden sökülüp, çıkarılır ve telleri gerekli yerlere yerleştirilir.

Yirmi yıldan beri her yıl, bilim adamları **Çip** parçalarını çoğaltarak, boyunu küçültmeyi

başardılar. İlk adım, bir kedinin bıyıkları inceliğindeki lehimlenmiş tellerle atıldı. Bu teller, bezelye büyüklüğündeki metal kutularda saklanan silikon kristallerinden elde ediliyordu. Sonra, üç ayaklı tabureye benzer ilk transistör yapıldı.

İlk transistör, **Bell telefon laboratuvarlarında**, 1974 yılında yapıldı. Bilindiği gibi ilk TV'lerde ve bilgisayarlarda, elektrik akımlarını kontrol etmek ve büyötmek için kocaman cam tüpler kullanılıyordu. Bu tüpler aşırı enerji çektiğinden ve sık sık yandığından yetersiz kalıyordu.

Oysa, transistörün de kusurları vardı. Teksas Instruments'dan **Jack Kilby** ve **Robert Noyce**, buna da çözüm yolu buldular. Böylece ilk **Çip** doğdu.

Çipte lehimlenmiş teller olmadığı için, hata oranı da çok düşük. Boyunun da çok küçük olması, elektrik sinyalleri düğmeden düğmeye geçerken, kısa bir yol katettiklerinden, zaman kazandırıyor. Dahası, 1000 transistörlü bir **Çip** eskiye oranla daha hızlı çalışabiliyor.

1990 yılında, mühendisler on milyon transistörü bir **Çipe** yerleştirmeyi tasarlıyorlar. Küçük bir bilgisayarda, 100 000 **Çip** bulunuyor. Bilgisayarlar, elektriği depo ederek, geçirerek ve kontrol ederek sayısız işi başarabiliyorlar. Bilgi-

sayılar, bizim dilimizden anlamadıkları için, sayılarımızı, harflerimizi ve sembollerimizi, küçük elektrik akımlarını temsil eden kodlar kullanıyoruz. Bilgisayarlar, yüksek voltajlı bir elektrik akımı, 1 hanesini temsil ediyor. Zayıf bir akım ise, 0 hanesini. Bilgisayarlar bilgiyi kodlamak için 0 ya da 1 sayısını kullanıyor. Bu düzen sadece iki sayıyı içerdiği için, bir bilgisayarın elektrik akımlarına, İngilizce "Binary digits" (ikili sistem) kelimelerinin kısaltılmasından oluşan "Bit" deniliyor.

Birçok bilgisayar, bilgiyi sekiz elektrik akımından oluşan kümeler halinde alıyor. Bayt adı verilen bu kümeler, sekiz tane "Bit"ten oluşuyor. Her bir "Bit", 0 ya da 1 ile temsil edildiğine göre, bu "Bit"ler 256 (2⁸) değişik şekilde sıraya konulabiliyor.

Verilen emirleri yerine getirebilmek için, bir bilgisayar "beynine" başvuruyor. Kişisel ev bilgisayarlarında, bu beyin tek bir Çipten oluşuyor. Tek başına bir Çip, tüm işlemleri yürütmeyeceği için, bir de verilen bilgiyi depo eden bellek Çipleri gerekli.

Japonya'nın Ticaret ve Endüstri Bakanlığı, MITI 1990 yılında, düşünebilen ve konuşabilen bir bilgisayar çeşitli dokümanları otomatikman çevirip yazabilecek. Verilen bilgiye göre, yargıya varacak ve kendi yaptığı hatalardan ders almasını bilecek.

Bilgisayarlardan, günümüzde, tıp alanında çokça yararlanılmakta. Artık verilere göre, bir bilgisayar hastalığı teşhis bile edebiliyor. ABD'nin Minnesota eyaletinde, 6 ile 18 yaş grubundaki tüm çocuklar, artık derslerini bilgisayar yardımıyla yapıyorlar. Bilgisayar, sınıfta öğretmenin yerini alarak, daha verimli bir şekilde çocuklara yeni yetenekler kazandırıyor. Eğitimcilere göre, böylece çocuklar küçük yaşta daha fazla öğrenmeye başlayacaklar.

Çiple atılan ilk adım, olağanüstü bilgisayarlar ve robotların yapılmasına neden oldu. Bugün Japonya'da futbol sahası büyüklüğünde bir fabrika, tamamen robotlar tarafından yönetiliyor.

Çipin yaratılmasıyla, hiç kuşkusuz yaşantımızda büyük bir devrim oldu. Gelecekte, Çip yepyeni alanlarda kullanılacak. Örneğin, sabah kalkınca, televizyonumuzu çevirip, gazeteyi ekranda okuyacağız. Her şey bir yana, yapay kol ve bacakların yapılması, ile gelecekte zekâmızı geliştirmek için beyinlerimize mikrodevrelerin yerleştirilmesi tasarlanıyor.

Gelecekte, Çip sosyal konulara da yeni boyutlar getirecek. Şimdi, yaşantımızı çalışarak aldığımız gelirle sürdürüyoruz. Gelecekte, robotların bütün işi yaptığı bir dünyada, gelir dağılımı nasıl olacak acaba?

Makineler düşünmeye başlarsa, insanlığına yapacak iş kalmayacak belki de. Her gelişme

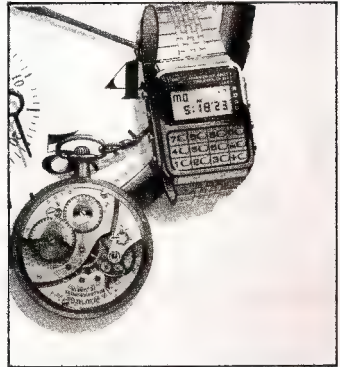
için, bir bedel ödemek zorundayız hiç kuşkusuz. Çiple birlikte yeni bir çağı başlattı bilim adamları. Bilgisayarların insanlığına yaptıkları hizmetler düşünülürse, ödememiz gereken bedel çok küçük.

BİLEĞİNİZDEKİ ÇİP

Çip, kol saatlerinden tık tıkları çıkartıp, yerine takvimleri, hesap makinelerini, müzik, oyun ve alarmı getirdi. Renklerle kodlanmış mekanik bir saate (1) ana zemberek, (mavi) tarafından depo edilen enerji, balans dişlisinin (portakal rengi) yinelenen hareketleri, dişliler (mor) ve kaçış dişlisi denen (portakal rengi) tarafından zaman ünitelerine dönüştürülüyor.

Kuvarz kristal saate (2) bu parçaların, karşı parçaları var. Piller, ana zembereğin yerine ana güç kaynağını oluşturuyor. Kaçış ve diğer dişlilerin yerine bir kuvarz kristali (3) kullanılıyor. Bu kristal, saniyede 32,768 gibi bir frekansla titreşim yapıyor. Devreler (4) bu titreşimleri on beş kez yarıya bölerek, bir saniyelik bölümler elde ediyorlar. Durdurulabilen saatlerde (5) 1, 024 frekanslı bir titreşim, bölünerek saniyenin onda ve yüzde biri elde ediliyor.

Çip aynı zamanda saatin göstergelerini de kontrol ediyor. (6) Bütün numaraları oluşturan elektrotlar arasında, milyonlarca sıvı kristal gidip geliyor. Elektrotlar, elektrik alanı oluşturarak, elektrikle yüklenince, kristaller saatin kadrantındaki polarıcı filmin üzerine toplanıp, kadranda gördüğümüz numara, gün, tarih, vs. yi oluşturuyorlar.



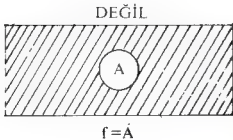
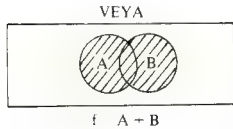
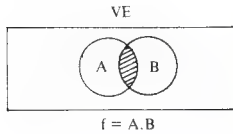
"Çip"lerin gelişmesiyle, bilgisayar teknolojisiyle çalışan, "dijital" saatler yaygınlaştı. (Üstte)

Doğruluk Tabloları

Boolean Cebirinde, üç değişik doğruluk tablosu kullanılır. Bunlar **standart doğruluk tablosu**, **koordinat tablosu** ve **Venn şeması**'dır. Biz burada Venn şemasını ve standart doğruluk tablosunu göstereceğiz.

Doğruluk tablolarında, bir denklemde ya da çizimde yer alan değişkenleri bütün olasılıklarıyla birlikte, fonksiyon çıkışlarının değişimleri gösterilir.

Venn Şeması: Venn şeması, Boolean Cebirinde pek kullanılmaz. Ama Boolean fonksiyonlarını çok açık bir şekilde gösterir. Bu nedenle herkesin bu şemalar üzerine en azından bir bilgi sahibi olması gereklidir.



Üstte üç temel fonksiyonun Venn şeması görülmektedir. Gözlemlenilmiş kısımlar, belirtilen fonksiyonu gösteriyor.

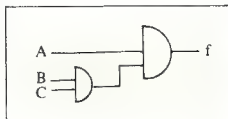
Standart Doğruluk Tablosu: Standart doğruluk tablosu, girişte yer alan elemanların olabilecek her türlü kombinasyonu göz önüne alınarak hazırlanır. Kombinasyon sayısını hesaplamak için girişte yer alan eleman sayısının 2'nin kuvveti olarak alır, sonucu çıkarırız. Bu, o sayıda giriş olan bir fonksiyonda yer alabilecek tüm kombinasyonların sayısını verir. Örneğin üç girişli bir fonksiyonda, başka bir deyişle üç elemanlı bir denklemde bulunacak bütün kombinasyonların sayısı $2^3 = 8$ 'dir. Eğer denklem dört elemanlıysa, kombinasyon sayısı $2^4 = 16$ 'ya çıkar. Bu hesaplamayı istediğimiz sayıya kadar uzatabiliriz.

A	B	C	f
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Üstte görülen şekilde üç elemanlı bir denklemin standart doğruluk tablosu yer alıyor. Bu doğruluk tablosuyla, her denklemin türlü kombinasyonlarında çıkacak olan fonksiyonları hesaplayabiliriz. Örnek olarak şöyle bir denklemi ele alalım:

$$f = A + (B^0 C)$$

Bu denklemin çizimi de şu şekilde olacaktır:



Bir de bunun standart doğruluk tablosunu çıkaralım.

A	B	C	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Standart doğruluk tablosu, bundan sonra göreceğimiz konularda bize çok yardımcı olacaktır. Boolean Cebirinin kurallarını incelerken, bu tablodan yararlanacağız.

Koordinat doğruluk tablosu, özellikle çok uzun ve karmaşık çizimleri, denklemleri sadeleştirmek için kullanılır. Bu şekilde bilgisayar devre şemalarında görülen karmaşık çizimler basite indirgenir ve işlevleri daha kolay anlaşılır. Ancak biz burada bilgisayar devre şemalarına girmeyeceğimiz için, koordinat doğruluk tablosunu da incelemeyeceğiz.

Boolean Cebirinin Kuralları

Boolean Cebiri, matematiksel her yöntem gibi kurallara sahiptir. Biz, burada bu kuralların bazılarını değineceğiz. Boolean Cebirinin normal cebirle birçok benzer yanları olmasına karşın, değişik yanları da vardır. Biz burada iki cebiri karşılaştırmayaacağız. Ama zaman zaman bazı kuralların ne kadar benzer, bazı kuralların da ne kadar farklı olduğunu görüp şaşıracağız.

Değişme Kuralları: Boolean Cebirinin bir toplama (VEYA fonksiyonu) için, bir de çarpma (VE fonksiyonu) için olmak üzere iki değişme kuralı vardır. Bu kural, fonksiyon içindeki değişkenlerin sırasının çıkışı etkilemediğini belirtir.

VEYA fonksiyonu için:

$$A + B = B + A$$

VE fonksiyonu için

$$A \cdot B = B \cdot A$$

Birleşme Kuralları: Birleşme kuralları, üç ve daha fazla değişken söz konusu olduğunda geçerlidir. Bu kural, işlemlerin istenilen sırada yapılabilmesini, değişkenlerin istenilen şekilde birleştirilebileceğini ve bunun çıkışı değiştirmeyeceğini belirtir.

VEYA fonksiyonu için:

$$A + (B + C) = C + (A + B)$$

VE fonksiyonu için:

$$A \cdot (B \cdot C) = C \cdot (A \cdot B)$$

Dağılma Kuralları: Dağılma kuralları, Boolean Cebiri ve normal cebirde görünüşte aynıdır. Ancak yorumlanışları biraz farklıdır. Biz bu kuralları, kapı çizimlerinden ve doğruluk tablolarından yararlanarak kanıtlayacağız.

Birinci kural:

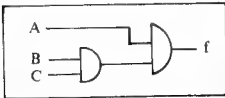
$$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$$

Bu kuralın doğruluğunu ispatlamak için, iki parçaya ayıracağız. Önce eşitliğin baş tarafını inceleyip, devre çizimini çıkarıp, ona göre bir doğruluk tablosu çıkaracağız. Sonra da aynı işlemleri bu kez eşitliğin sonuna uygulayacağız. Eğer doğruluk tabloları birbirinin eşi çıkarsa, kuralımız kanıtlanmış olacaktır.

Önce eşitliğin baş tarafını inceleyelim:

$$A \cdot (B + C) f$$

Bu denklemin devre çizimi şöyle olacaktır:



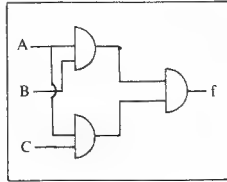
Şimdi de doğruluk tablosunu çıkaralım. Denkleminde üç değiş-

ken eleman olduğu için, $2^3 = 8$ olasılığı vardır. Buna göre, tabloyu şöyle kurarız.

A	B	C	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Sonra da VE ve VEYA kapılarının genel kurallarına göre, fonksiyon çıkışları belirlenmiş oluyor. Şimdi de aynı işlemleri eşitliğin diğer tarafına uygulayalım.

$$f = (A \cdot B) + (A \cdot C)$$



Bu denkleminde A değişkeni iki kez kullanılıyor. Ama bunun için çizime iki tane A girişi yapmamız gerekmez. Daha önce kullanılan girişten bir kol daha çıkartıp, ikinci kapıda kullanabiliriz. Ancak bu arada çıkardığımız kol, B girişinin üzerinden geçiyor. Bunu belirtmek için tam iki kolun kesiştiği yerde kolların birisini yarım daire şeklinde çiziyoruz. Bu kural, bütün devre çizimlerinde karışıklığı önlemek amacıyla kullanılır. Eğer iki kol birbirini dümdüz keserse, bu, devrede ikisi birleşiyor demektir ki, bizim çizimimize göre yanlış olur.

Şimdi doğruluk tablosunu çıkaralım:

A	B	C	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

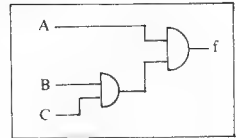
İki denklemin fonksiyon çıkışlarının aynı olduğu görülüyor. Bu şekilde, eşitliğin de doğruluğunu ispatlamış oluyoruz.

İkinci kural:

$$A (B \cdot C) (A \cdot B) (A \cdot C)$$

Bu kuralı da daha öncekiler gibi kanıtlayalım.

$$f = A + (B \cdot C)$$

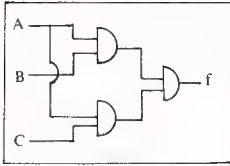


Devre çizimi

A	B	C	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Doğruluk tablosu

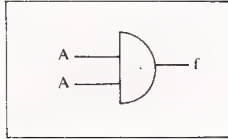
$$f = (A + B) \cdot (A + C)$$



Devre çizimi

$f = 0 + 0$ yani '0'dır, eğer $A + 1$ ise, sonuç $1 + 1 = 1$ 'dir.

devre



Devre çizimi

A	B	f
0	0	0
0	1	0
1	0	1
1	1	1

A	B	C	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

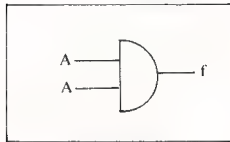
Doğruluk tablosu

Bu iki tablonun da fonksiyon çakışları aynı olduğuna göre, eşitlik kanıtlanmış oluyor.

Totoloji Kuralları: Birinci kural:

$$A \cdot A = A$$

Bu kuralın kanıtlanması oldukça kolaydır. VE fonksiyonunda bir eleman eğer '0' ise, $A = 1$ ise çıkış '1'dir.



Devre çizimi

İkinci kural:

$$A + A = A$$

Bu kuralın kanıtlanması da birincisine benzer. Eğer $A = 0$ ise,

Tümlene Kuralları: Birinci kural:

$$A = 0, \bar{A} = 1$$

$$A = 1, \bar{A} = 0$$

İkinci kural:

$$A \cdot \bar{A} = 0$$

Üçüncü kural:

$$A + \bar{A} = 1$$

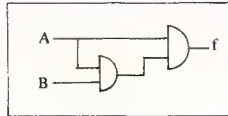
Çift kural:

$$\bar{\bar{A}} = A$$

Soğurma Kuralları: Birinci kural:

$$A \cdot (B + B) =$$

Bu denklemin devre çizimine ve doğruluk tablosuna bakalım.
 $f = A \cdot (A + B)$



Devre çizimi:

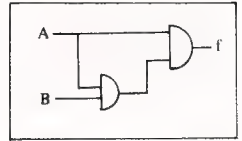
A	B	f
0	0	0
0	1	0
1	0	1
1	1	1

Doğruluk tablosunda da görüldüğü gibi çıkış fonksiyonu ile A değerleri aynıdır.

İkinci kural:

$$A + (A \cdot B) = A$$

Bu kuralı da aynı şekilde kanıtlayabiliriz.
 $f = A + (A \cdot B)$



Devre çizimi

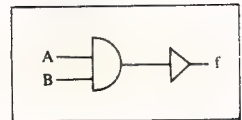
Yine çıkış fonksiyonlarıyla A'nın değerlerini karşılaştırsak, ikisinin de aynı olduğunu görürüz.

De Morgan Teoremi: Birinci kural:

$$\overline{A \cdot B} = \bar{A} + \bar{B}$$

Kanıtlamak için devre çizimlerini ve doğruluk tablolarını karşılaştıralım.

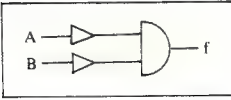
$$f = \overline{A \cdot B}$$



Devre çizimi

A	B	f
0	0	1
0	1	1
1	0	1
1	1	0

$$f = \overline{A+B}$$



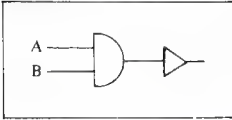
Devre çizimi

A	B	f
0	0	1
0	1	1
1	0	1
1	1	0

İkinci kural:

$$\overline{A+B} = \overline{A} \cdot \overline{B}$$

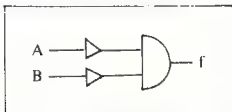
$$f = \overline{A+B}$$



Devre çizimi

A	B	f
0	0	1
0	1	1
1	0	1
1	1	0

$$f = \overline{A \cdot B}$$



Devre çizimi

Sayı Düzenlerini Çevirme

Bir sayı düzenindeki sayıları, başka bir düzene çevirebiliriz. Yalnız doğaldır ki, kurallara uygun olarak... Bir düzeni "temel düzen" olarak almamız gerekiyor. Biz, çok yakından tanıdığımız 10'luk düzeni kullanacağız.

Çevirilerde, tamsayıları ayrı, virgüllerin sağındaki sayıları ayrı ayrı ele alacağız. Çünkü, her iki tarafa da ayrı ayrı işlemler uygulanır.

Herhangi Bir Sayı Sistemini Onluk Düzene Çevirme: Daha önce ikilik düzeni anlatırken gör-

müştük: Sayıların her basamağı, 2'nin bir kuvvetini temsil eder. Örneğin, 11 sayısı, 2'lik düzende 1

$\times 2 + 1 \times 2 = 3$ 'tür. Aslında biz bu işlemle, 2'lik düzende bir sayıyı 10'luk düzene çevirmiş olduk. Ama daha açık bir şekilde bakarsak;

$$10110 = 1 \times 2 + 0 \times 2 + 1 \times 2 + 1 \times 2 + 0 \times 2 = 22 \text{ 'dir.}$$



Bilgisayar kullanımında en çok kullanılan düzen 10'luk düzendir. Bu düzen, kolaylıkla 2'lik düzene çevrilebildiği için, donanım açısından da bir sorun

yaratmaz. Bildiğiniz gibi bilgisayarın iç yapısına **donanım**, işletimine de **kullanım** denir.

Onaltılık düzende, 0 dahil olmak üzere 16 rakam vardır. 9'dan sonraki rakamlar A, B, C, D, E, F olup sırasıyla 10, 11, 12, 13, 14, 15'i temsil eder.

Onaltılık düzen, daha sonra bilgisayar kullanımı konularına geçtiğimizde bize çok yardımcı olacaktır.

Çevirme işlemi için genel bir kural hazırlarsak:

1. Önce, sayının basamakları sayılıp ona göre o sayı düzenine göre, hangi basamağın kaçını kuvveti temsil edeceği belirlenir.

2. Her bir basamaktaki rakamlar, düzende sayının kaçını kuvvetini temsil ediyorsa, onlarla çarpılır.

3. Çarpımlar toplanır.

Onluk Düzendeki Başka Düzenlere Çevirme: Bu tip çevirmede de şöyle bir yol izleriz:

1. Onluk düzende sayı, hangi düzene çevrilecekse: O sayıya bölünür.

2. Eğer kalan olmazsa, kalanlar bölümüne bir 0 yazılır. (Örneklere bakınız.) Eğer kalan varsa, kalanlar bölümüne o sayı yazılır.

3. Bölüm, yeniden o sayıya bölünür ve yani işlem tekrarlanır.

4. Bölüm, bölen sayıdan daha küçük oluncaya kadar, işlem tekrarlanır. Bölümdeki en son sayı da, son olarak kalanlar bölümüne yazılır.

Şimdi birkaç örnek yapalım: 65 sayısını 10'luk düzenden 2'lik düzene çevirelim.

ilk adım	ikinci adım
$\begin{array}{r} 65 \overline{) 7} \\ \underline{63} \\ 2 \end{array}$	$\begin{array}{r} 9 \overline{) 7} \\ \underline{7} \\ 2 \end{array}$
2 kalanlar	2 1

1 sayısı, 7'den küçük olduğu için, bölme işlemini durdurup, onu da kalanlar bölümüne yazıyoruz. Yalnız kalanlar bölümündeki rakamlar, bize gerçek sayının tam tersini verir. O yüzden, sonucu yazmadan önce, sayıları ters çevirmemiz gerekir.

Yani 65 sayısının 2'lik düzendeki eşdeğeri, 122'dir.

Bir örnek daha yapalım. Bu kez de 1707 sayısını 16'lık düzende yazalım:

$\begin{array}{r} 1707 \overline{) 16} \\ \underline{16} \\ 106 \end{array}$	$\begin{array}{r} 105 \overline{) 6} \\ \underline{105} \\ 0 \end{array}$
kalanlar	11 10 6

Yani 1707 sayısı 16'lık düzende 6 AB'ye eşit oluyor. Kalanlar bölümündeki 11 ve 10 sayılarını, 16'lık düzende aynı şekilde yazamayacağımız için, eşdeğer harflere çeviriyoruz.

İkili Düzene İlişkili Düzenler: Birbirleriyle ilişkili olan bazı düzenleri birbirine çevirmek çok basittir. Örneğin 4'lük düzen, 8'lik düzen ve 16'lık düzen 2'nin kuvvetleri olduğu için, 2'lik düzende ilişkili düzenlerdir. Bunu örneklerle çok daha iyi anlayacağız. Önce 8'lik düzende bir sayıyı, 2'lik düzene çevirelim. Sekiz, ikinin üçüncü kuvvetine eşit olduğu için, 8'lik düzende her basamak, 2'lik düzende üç basamakla ifade edilir. Bunu gösterebilmek için bir sayı ele alalım:

1 5 4 5 1 8'lik düzen
002 101 100 101 001 2'lik düzen

Bu şekilde 8'lik düzende 15451 sayısının 2'lik düzende eşdeğeri olan 001101100101001 sayısını bulmuş oluyoruz.

Onaltılık düzende A 572 sayısı, ikili düzende 101001010110010 sayısına eşit oluyor.

İkili düzende bir sayıyı dörtlük, sekizlik ya da onaltılık düzene çevirirken de bütün bu işlemlerin tersini yapıyoruz.

0001 1011 0010 1001 2'lik düzen
B 2 9 16'lık düzen

Yukarıda görülen şekilde, ikili düzende hangi sayıya çevrilecekse, o sayının ikinin kaçımı kuvvetiye o kadar basamaklık parçalara ayrılır. Sonra da parça parça eşdeğer sayılar bulunup bir araya getirilir.

Virgölün Sağındaki Basamakların Çevirimi: Çevirim yaparken getirilen bu ayırım, bilgisayar için yapılması çok güç olan bazı problemleri kolaylaştırır. Çünkü bazı sayıların mutlak çevirimleri olanaksızdır. Siz de "bayağı kesirleri" "ondalık sayılara" çevirmeye çalışırken, bu sorunla sık sık karşılaşacaksınız. İki tip ondalık kesir vardır. Sonlu kesirler, ki bunların sonuçları mutlak ve sonsuz kesirler. Sonsuz kesirlerde kaç basamak eklenirse eklenir, mutlak bir değer elde edilemez.

Sonlu kesirlerle örnek olarak verebileceğiz:

$$\frac{1}{2} = 0,5 \quad \frac{1}{5} = 0,2 \quad \frac{1}{10} = 0,1 \quad \frac{1}{16} = 0,0625$$

Sonsuz kesirlere örnek olarak:

$$\frac{1}{3} = 0,333 \dots \quad \frac{1}{6} = 0,1666\dots \quad \frac{1}{13} = 0,769230 \text{ } 7692\dots 2^3$$

Sonsuz kesirleri çevirirken, sayının mutlak değere ne kadar yakın olmasını istiyorsak, basamakları o kadar uzatırız.

Biz bu çevirme işlemini yaparken, ikili düzeni örnek olarak alacağız. Diğer düzenlerdeki çevirimler de bu yöntemlere göre yapıldığı için, onlara da ayrıca girmenin pek bir yararı olmaz.

Örnek olarak ikili düzende 0.11 sayısını onluk düzene çevirelim:

İkili Düzende	
0.1	=
+ 0.01	=
0.11	
↓	
Bayağı Kesir	
1/2	=
+ 1/4	=
3/4	
↓	
Ondalık Kesir	
0.5	
+ 0.25	
0.75	

Bu yöntemin de genel kuralı şudur:

1. Virgülün sağındaki her basamak sağa doğru gittikçe, sayı düzenine göre, o sayının kuvvetinin bir bölümünü temsil eder. Yani ikilik düzende virgülün hemen yanındaki basamak $1/2^1 = 1/2$ ikinci basamak da $1/2^2 = 1/4$ temsil eder.

2. Her basamaktaki rakam, o basamağın gösterdiği kuvvet derecesinin bir bölümüyle çarpılır. Başka bir deyişle, rakam o kuvvete bölünür.

3. Her rakam için yeni bir değer elde edildikten sonra, bunlar toplanır.

Bununla ilgili bir de örnek çözelim: 0,125 sayısını ikilik düzene göre yazalım:

$$\begin{array}{r} - 0,125 \\ \times 2 \\ \hline 0 \leftarrow 0,250 \\ \times 2 \\ \hline 0 \leftarrow 0,500 \\ \times 2 \\ \hline 1 \leftarrow 1,000 \end{array}$$

• Yani 0,125 sayısının ikilik düzendeki eşdeğeri 0,001'dir.

olduğu gibi toplanır. Eğer toplam 2'den küçükse, aynen yazılır. Yani elde kalan bir şey yoktur.

2. Eğer toplam 2'ye eşit ya da büyükse, 2'ye bölünür. Bölüm, elde var olan sayıdır. Kalan sayı, o basamaktaki sonuç bölümüne yazılır.

Bu konuyla ilgili bir örnek yapalım.

$$\begin{array}{r} 111 \\ 101 \\ 001 \\ + \\ \hline 1101 \end{array}$$

$$0,10101 = 1/2^1 + 0/2^2 + 1/2^3 + 0/2^4 + 1/2^5 = 1/2 + 1/8 + 1/32 = 21/32 = 0,65625$$



Bir örnek daha yapalım. Bu kez de 0.10101 sayısını onluk düzene çevirelim. (üstte)

Onluk düzendeki bir kesiri başka bir düzene çevirmek içinse, şu kuralları geçerlidir:

1. Tüm kesir, hangi düzen kullanılacaksa o sayıyla çarpılır. Biz örneklerimizde ikilik düzeni kullandığımız için, bu sayı 2 olacaktır.

2. Eğer bir tamsayı çıkarsa (kesirle birlikte ya da kesirsiz), o sayı bir tarafa yazılır. Eğer tamsayı çıkmazsa 0 yazılır.

3. Eğer kesir kaldıysa aynı işlem yenilenir. Bu iş, artık kesir kalmayana dek sürdürülür.

4. Kesirli sayılar bitince işlem durdurulur.

Bilgisayar Ve Aritmetik

Bir bilgisayarın düşünme yeteneği yoktur. Verilen problemleri çok basit birtakım yollarla çözer. Bütün bu işlemler de, **Boolean aritmetiği** kuralları içinde yapılır. Bu bölümde bir bilgisayarın çalışma mantığını çözecek, işlemleri ona benzer bir şekilde yapmaya çalışacağız. Biz insan olduğumuz için tam onun gibi yapmamıza olanak yok, biz birtakım kuralları kendi mantığımıza göre ayarlayacağız. Ama bunun bize bilgisayar mantığını kavramamız açısından bir sorun yaratmayacağı gibi, bazı kolaylıklar da sağlayacağını düşünüyoruz.

Toplama: Toplama işleminin genel kuralları, bizim yaptığımız toplama işleminin kurallarıyla aynıdır. Yalnız her şey **ikilik düzene** göre gerçekleştirilir.

Şöyle:

1. Sayılar onluk düzende

Sağdan ilk basamağı toptarsak, elimizdeki sayı 10'luk düzende 3 oluyor. Bu sayı 2'den büyüktür. Böldüğümüzde, bölüm 1, elde olan sayıdır. Kalan 1 ise, o basamağın sonuç bölümüne yazılır. İkinci basamağa geçtiğimizde, eldeki 1 sayısıyla diğer 1 sayısını topluyoruz. Bu da 10'luk düzene göre, 2'ye eşit oluyor. Bu nedenle ki, bir kez de 2'yi kendine bölüyoruz. Bölüm 1, elde var olan sayıdır. Kalan 0 ise, o basamağa yazılır. Son basamakta ise, elimizdeki 1 sayısıyla diğer 1'leri topluyoruz. Yine 10'luk düzende bu, 3 ediyor. Onu da aynen başta yaptığımız gibi hesaplayıp, 1'i yazıyoruz. Ancak toplama işlemi bittiği için elde sayı bırakmamıza gerek kalmıyor. Son sayıyı aynen yazıyoruz.

Gördüğümüz gibi toplama işlemi, bizim her gün yaptığımız toplama işlemine çok benziyor. Yalnız biz 10'luk düzeni kullandığımız için, toplam 10'a eşit ya da büyük olunca eldeki sayıyı hesaplıyoruz. Burada aynı şeyi toplam 2'ye eşit ya da büyük olunca yapıyoruz.

Eğer elimizdeki sayı yine 2'den büyükse, o zaman bir sonraki basamağa geçiyoruz.

Yani:

$$\begin{array}{r}
100 \\
100 \\
10001 \\
10101 \\
10111 \\
11011 \\
+ \\
\hline
1011000
\end{array}$$

Örneğin yukarıdaki işlemi yaparken, ilk basamaktaki toplam, 10'luk düzene göre 4 oluyor. Bu sayının 2'ye bölümü yine 2 olduğu için, bunu bir sonraki basamağa yazmamız olanaksızlaşıyor. İkinci bir bölme daha yapıp, sonucu üçüncü basamağa yerleştiriyor; ikinci basamağın yerine de 0 yazıyoruz. İkinci basamaktaki toplam da 2 olduğu için, elimizdeki ikinci sayıyı da üçüncü basamağın üzerine yerleştiriyoruz. Böylece üçüncü basamakta dört tane 1 oluyor. Bunun toplamını da aynen baştaki gibi yapıp işleme devam ediyoruz. Son basamağa gelince, elimizde beş tane 1 oluyor. Artık geçirecek başka bir basamak kalmadığı için, 5 sayısını aynen 2'lik düzendeği gibi yazıyoruz. Böylece sonucu bulmuş oluyoruz.

Çarpma: Çarpma işleminin kuralları da, normal çarpma işlemine benzer.

1. Sayılar, onluk düzendeği gibi yapılır.

2. Eğer çarpılacak sayılar iki ya da daha fazla basamaklıysa, normal bir çarpma işlemindeki gibi her basamakla yapılan işlemler, birer kaydırılarak alt alta yazılır, sonra da toplanır.

3. Toplama işlemi, yukarıdaki gibi yapılır.

Şimdi de bununla ilgili bir örnek çözelim:

$$\begin{array}{r}
1100 \\
1011 \\
\times \\
\hline
1100 \\
1100 \\
0000 \\
1100 \\
+ \\
\hline
10000100
\end{array}$$



Günümüzün bilgisayarlarından biri Norsk-Data ND 100. Bilgisayara çok kısa bir süre içinde çeşitli birimler eklenerek, gelişmiş bir sisteme dönüştürülebiliyor (Üstte)

Örnekten de açıkça görüldüğü gibi, çarpma işleminin en zor olan kısmı, basamak basamak çarpımları toplamaktır. Çünkü her sayının çarpımı ya 1, ya da 0'dır.

Tümleme Toplamı Yoluyla Çıkarma İşlemi: Matematikte yapılan tüm işlemlerin, sadece toplama işlemi ile gerçekleştirilmesi olasıdır. Mademki böyle, matematikçiler niye durmadan yeni yöntemler geliştiriyor diyebilirsiniz. Bunun cevabı çok basit: Zaman kazanmak için. Örneğin 125 sayısının karesi, çarpma işlemiyle epey kısa bir zamanda çıkarılabilir. Eğer zamanın önemi yoksa, aynı sonuca bir dizi toplama işleminden sonra da varılabilir. Bunun için 125'i 125'e yuzyirmibeş kez eklememiz yeterlidir. Ancak bunun için harcanan zaman içinde daha pek çok işlem yapılabilir. Bizim verdiğimiz çok basit bir örnekti. Biraz sabırla sonuç çıkarılabilirdi. Ancak çok daha karmaşık ve uzun işlemlerde, sadece toplama yoluyla sonuca ulaşabilmek yılları alabilir.

Günümüzde, bilgisayarın bize sağladığı en büyük avantaj, çabukluktur. Bunu bir örnek ile göstereyim.

Yirmi yaşına gelmiş birini ele alalım. Bu kişi matematik öğreniyor ve bir probleme başlıyor. Söz-gelimi, bir problemi çözmek, bu kişinin 50 yılı alacak. Yemek, uyku ve diğer gereksinimleri de içinde olmak üzere, beş saniyede bir işlem yapan bir matematikçi için; 50 yıl çok uzun bir süredir. Günde 24 saatten yılda 365 güne, bu kişi 50 yıl içinde ortalama 315 milyon işlem yapabilir.

Şimdi başka bir matematikçi düşünelim. Bu kişinin de, saniyede 1 milyon işlem yapabilen bir bilgisayarı olduğunu varsayalım (bu ne en yavaş, ne de en çabuk işlem yapan bir bilgisayardır). İkinci adamımız, bilgisayarının yardımıyla aynı işlemi beş dakikadan biraz fazla bir surede çözebilir.

Bu örneği biraz daha geliştirsek, ilk matematikçi hayatı boyunca dinlenmeye vakit bulmadan sadece birtak problem çözmüş oluyor. Diğerisi ise, aynı problemi çözdükten sonra, daha binlerce problem çözmeye zaman bulabiliyor. Bu kişinin 30 yıl böyle çalışabileceğini düşünecek olursak, bu süre içinde birincisi

gibi çalışan 360 matematikçinin yaptığını yapabiliyor.

Bu örnek, bilgisayarın bütün işlemlerle toplama yapmasının pek bir zaman kaybı olmadığını gösteriyor. Bilgisayar, en karmaşık problemleri bile bu yöntemle çözüyor.

Çıkartma işlemi de iki ya da üç adımda gerçekleştiriliyor. Önce tümleyeni alınıyor, daha sonra da çıkartılacak olan sayıyla toplanıyor. Daha başka bir yol da tümleyeni 1 ile toplayıp, sonra çıkartılacak sayıyla topluyor. Bu düzen, normal çıkartma işleminde hemen hiç kullanılmaz. Ancak ikilik düzende bir sayının tümleyeni almak çok kolaydır. Tüm 0'lar 1, tüm 1'ler 0 yapılır.

Buna bir örnek verelim. 110111011 sayısının tümleyeni 001000100'dür.

Tümleme yoluyla çıkartma işlemi şu şekilde yapılır:

1. Çıkarılacak sayının tümleyeni alınır.

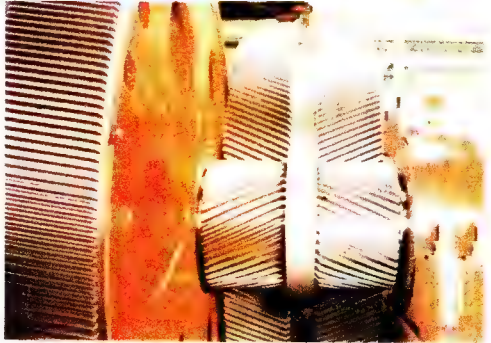
2. Çıkartılan sayıyla eklenir.

3. Eğer bu toplama işleminin sonras, eğer fazladan bir basamak çıkarsa, bu o sayının pozitif bir sayı olduğunu gösterir. Eğer yoksa, sonuç negatifdir ve bu sayının yeniden tümleyeninin alınması gerekir.

Bunu bir örnekle gösterelim. 10110 sayısını 11001 sayısından çıkaralım. Önce 10110 sayısının tümleyenini almamız gerekiyor. Bunun için de 1'leri 0, 0'ları da 1 yaparız. Sonuç 01001'dir. Şimdi de bu sayıyla 11001'i toplayalım.

Fabrikalara yeni birimler, yeni üniteler de bilgisayarların yardımıyla ekleniyor. (En üstte)

Bilgisayarlar, endüstride de çoklukla kullanılıyor. Ürünlerin kalitesi, onların yardımıyla kontrol ediliyor. (Ortada)



$$\begin{array}{r} 11001 \\ + 01001 \\ \hline 1\ 00010 \end{array}$$

Elimizde fazladan bir basamak var. Yani beş basamaklı iki sayıyı topladık ve altı basamaklı bir sayı elde ettik. Bu basamak, sonucun pozitif olduğunu gösteriyor. Yani, yeniden tümleyenini almamız gerekiyor.

Çıkan sayıya, bu 1 sayısını da ekliyoruz.

$$\begin{array}{r} 10 \\ + 1 \\ \hline 11 \end{array}$$

Bu çıkartma işleminin sonucu 11'dir. Sağlamasını yapmak için, bu sayıyı 10110 sayısıyla toplayalım.

$$\begin{array}{r} 10110 \\ + 11 \\ \hline 11001 \end{array}$$

Bu şekilde yaptığımız işlemin de doğruluğunu kanıtlamış olduk.

Şimdi başka bir örnek gözelim.

$$\begin{array}{r} 10011 \\ - 10101 \\ \hline ? \end{array}$$

10101 sayısının tümleyenini 01010'dur. Bu sayıyla ilk sayıyı toplarsak:

$$\begin{array}{r} 10011 \\ + 01011 \\ \hline 11101 \end{array}$$

Büyük şirketler, muhasebe hesaplarını bilgisayarların yardımıyla yaparlar. Böylece, büyük bir kadronun, uzun bir sürede yapacağı bu işlemler, kısa bir süre içinde gerçekleştirilmiş olur. (Üstte)

Bu toplamda fazladan bir basamak çıkmadı. Bu çıkan sayının negatif olduğunu gösteriyor. Yani, bu çıkarımın asıl değerini bulmak için toplamın da tümleyenini almamız gerekir. Doğal olarak bu arada da başına eksi işaretini koyuyoruz. Çıkartma işlemini yeniden yazacak olursak:

$$\begin{array}{r} 10011 \\ - 10101 \\ \hline -10 \end{array}$$

Bölme: Bölme işlemi, bilgisayarlarda bir dizi çıkartma işlemi ile gerçekleştirilir. Bölen, sonuç 0 ya da negatif bir sayı çıkana kadar bölünenden çıkarılır. Negatif sonuç, bilgisayara gereğinden fazla çıkartma yaptığını gösterir.

Çıkartma işlemlerinin kaç kez yapıldığını belirleme işini bir sayaç üstlenir. Negatif bir sayı çıkar çıkmaz da otomatikman kendi tuttuğu sayıdan bir çıkarır. Eğer çıkan sayı 0 olursa, bu, sayının kendisinden bir çıkartmasını engeller.

Sonuçta bölme işlemi tamamlanmış, sayaçta bölüm tam sayı halinde bulunmaktadır. Eğer kalan varsa, onu da son çıkartma işleminden sonra kalan sayı gösterir.

Bölme işlemindeki çıkartma-





Gazetelerin baskı makineleri de bilgisayar kontrolüyle çalışır. Bu karmaşık makineler, yalnızca bir kontrol tablosuyla yönetilir. (En üstte)

Bilgisayar kontrollü bir film banyo makinesi. (Altta)

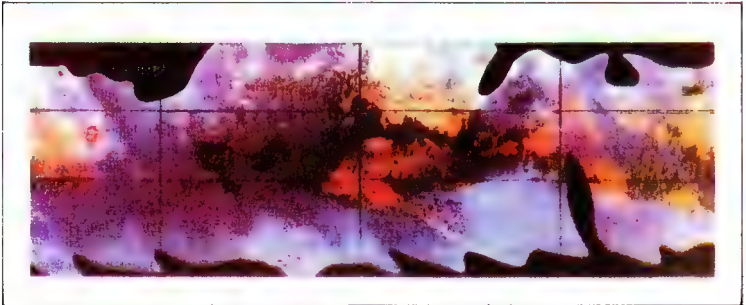
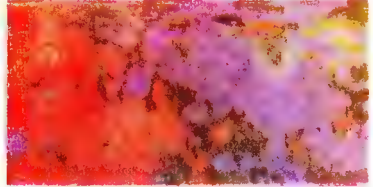
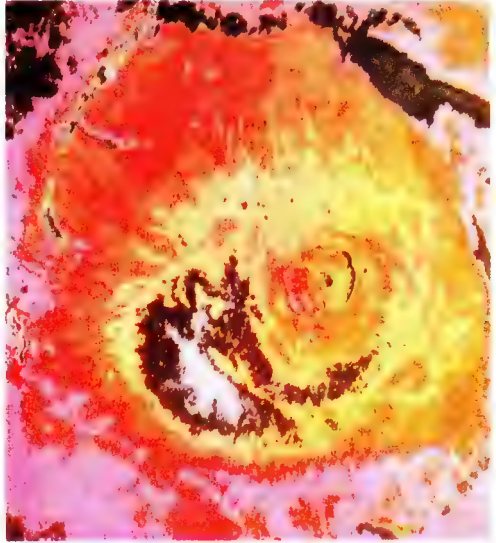
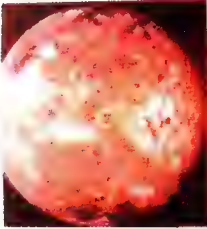
lar da, yine tümleyen metoduyla yapılır.

Buna da bir örnek verelim:

$$\begin{array}{r} 1000 \quad | \quad 101 \\ \hline \quad \quad ? \end{array}$$

Önce 101 sayısının tümleyeni alalım. 0'ları ve 1'leri değiştirsek, sonuç 1010 çıkıyor. Şu konuya dikkat etmemiz gerekir. Eğer çıkarılan sayı çıkan sayıdan daha fazla basamağa sahipse, çıkarılan sayıyı da öyle düşünme-

Teleskoplar, çok uzaklarda bulunan gezegenleri yakınıma getirir. Özel fotoğraf makinelerince resimleri çekilen bu gezegenlerin filmleri, bilgisayarlar tarafından düzene sokulur. Böylece, o gezegendeki ılık koşulları saptanmaya çalışılır. (Yanda)



miz gerekir. Çıkan sayıyı da o kadar basamağa sahipmiş gibi düşünüp, tümleyenini ona göre almamız gerekir. Yani 101 sayısı üç basamaklı. Ama biz bunu dört basamaklı olarak düşünüyoruz. Doğaldır ki bu, sayının değerini hiçbir zaman değiştirmek anlamına gelmez. Başka bir deyişle 101 sayısını dört basamaklı olarak düşünürsek, bu bizim onu 0101 olarak düşünmemiz demektir. Bu sayının da tümleyeni 1010 olur. Şimdi toplama yapmaya geçelim

$$\begin{array}{r} 1000 \\ 1010 \\ + \\ \hline 1\ 0010 \\ \swarrow 1 \\ 0101 \end{array}$$

İlk işlemden çıkan sonuç 101'dir. Bu sayı pozitif bir sayı olduğu için çıkartmaya, yani daha doğru bir şekilde söylersek bölmeye devam edebiliriz.

$$\begin{array}{r} 0101 \\ 1010 \\ + \\ \hline 1111 \end{array}$$

Bu kez, fazladan bir basamak çıkmadı. Bunun tümleyenini alırsak 0000 çıkıyor. Yani bölme işleminin sonucunda kalan sayı 0'dır. Bu arada sayacımız da ikide durdu. Demek ki, 1000 sayısının 101'e bölümü 2'ymiş. Biz bu sayıyı da ikilik düzene çevirirsek, bölme işlemini yeniden şu şekilde yazabiliriz:

$$\begin{array}{r} 1000 \quad | \quad 101 \\ \hline 10 \end{array}$$

Bu dört işlem, bilgisayarın yaptığı en temel işlemlerdir. Kendisine verilen problemleri en çok bu işlemleri kullanarak, daha



Günümüz bilgisayar teknolojisinin temel taşlarından biri: Çip'ler. (Üsste)

doğrusu sadece ve sadece toplama işlemini kullanarak yapar. Diğer işlemlerin de, çıkartma, bölme, çarpma işlemlerinde olduğu gibi özel bir yöntemi vardır. Bilgisayarlar, hiçbir işlemi bizim anladığımız, yaptığımız şekilde yapmazlar.

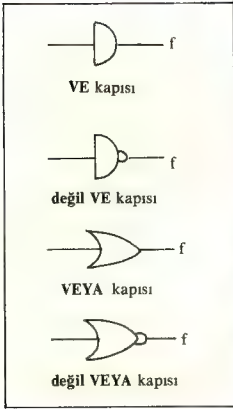
Flip-Flop

Bilgisayarlar, **flip-flop** adı verilen bir kapı düzeni aracılı-

ıyla işlemleri yaparlar. Bu düzenlerin özelliği, kendilerine başka bir uyarı gelene kadar daha önce verilmiş olan bilgiyi saklamalarıdır. Yani, bir başka uyarı gelmezse, elindeki bilgiyi sonsuza kadar saklayabilir. Biz burada size bir **flip-flop**un elindeki bilgiyi nasıl sakladığını göstermeye çalışacağız. Bu saklanan bilgiler olmadan bilgisayar, hiçbir problemi çözemez.

Değil VEYA, ve Değil VE Kapıları: Daha önce VEYA, VE ile DEĞİL kapılarını görmüştük. DEĞİL ve VEYA kapısı ya da DEĞİL ile VE kapısı, çok kulla-

nılan kapılar olduğu için; birlikte de gösterilebilirler.



Bir sistemin yazıcı ünitesi.
(Ustte)

Bir de fonksiyonlarını inceleyelim:

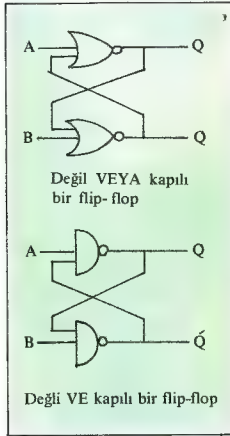
A	B	f
0	0	1
0	1	1
1	0	XI
1	1	0

değil VE

A	B	f
0	0	XI
0	1	0
1	0	0
1	1	0

değil VEYA

kapısından meydana gelir. Bu kapıların her birinin çıkışı, diğerinin girişini tamamlar. Bunu çizimleriyle gösterelim:



Görüldüğü gibi değil VE kapısı VE'nin tam tersi, değil VEYA kapısı da VEYA'nın tam tersi bir fonksiyon çıkarıyor.

Flip-Flop: Flip-floplar, iki değil VE, ya da iki değil VEYA

Bunların doğruluk tabloları da şu şekildedir:

A	B	Q	Q̄
1	0	1	0
0	0	1	0 (A=1, B=0'dan sonra)
0	1	0	1
0	0	1	0 (A=0, B=1'den sonra)
1	1	0	0

değil VEYA kapılı flip-flopun doğruluk tablosu.

A	B	Q	Q̄
1	0	0	1
1	1	0	1 (A=1, B=0'dan sonra)
0	1	1	0 (A=0, B=1'den sonra)
1	1	1	0
0	0	1	1

değil VE kapılı flip-flopun doğruluk tablosu

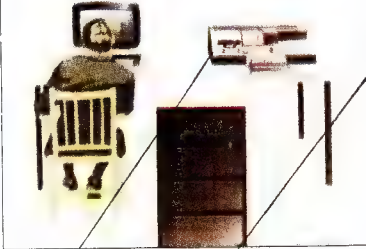
Bu kapıların her birinin girişi bir diğerinin çıkışını tamamladığından, içindeki bilgiyi sürekli yeniler. Bu şekilde kendilerine verilen bilgiyi, başka bir uyarı gelmediği sürece sonsuza kadar saklar.

Daha başka şekillerde, başka türlü kapılardan oluşmuş **flip-flop** da vardır.

Bir Kompüter Sistem

Terminal Odası

Bilgisayar Merkez Odası



Kart Okuyucu Veriler kartların üzerine delinir. okuyucu onları elektrik sel uyarılara çevirir

Elektronik Bağlantı Merkez odasıyla, uzakta terminaler arasındaki bağlantı, telefon tellerine sağlanır.

Manyetik Disk Belleği Veriler, önceden çıkarılıp saklanabilen diskler üzerine yazılır

Satır Yazıcısı Çıkış: bir saniyede birçok satır basan yazıcıyla yapılır

Manyetik Teyp Belleği Bu teypler evde kullanıdıklarıımızza benzer. Ancak çok daha hızlı çalışırlar

Bilgisayar Donanımı

BİLGİSAYAR da, bir araba gibi ufak parçalar - dan oluşur. Küçük bir bilgisayar da bütün bu parçalar birliktedir. Yani ayrı ayrı çalıştıkları halde tek bir muhafaza içinde saklanır. Büyük bilgisayarlarda ise, her parça ayrı ayrı muhafazalardadır. Bu parçalar bir odaya da konabilir, ayrı odalarda, ayrı binalarda, hatta ayrı ülkelerde bile olabilir.

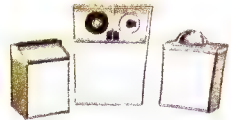
Bilgisayarın Kalbi: Bilgisayarların kalbi, **Merkezi İşlem Ünitesi**'dir. (Central Processing Unit, CPU). Bu üniteye işlemci adı verilir, çiplerden oluşmuş bir bölüm

vardır. Bütün işlemler burada yapılır. Ünitenin içindeki diğer bölümler işlemciye bu konuda yardım ederler.

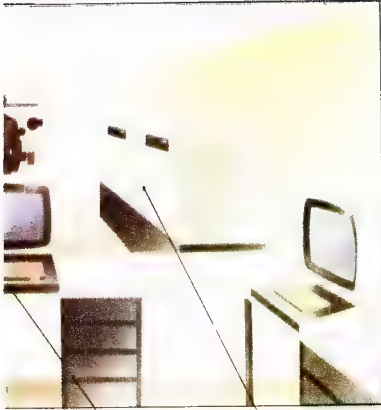
İşlemcilerin iki ortak özellikleri vardır. Bunlardan birincisi, kendisine verilen bilgileri zamanında alabilmek, ikincisi de sonradan gerekli olabilecek bilgilerin bir yerde saklanmasını sağlayabilmektedir. İşlemcinin bu özellikleri yerine getirebilmesi için, CPU içindeki belleğe kul-

lanacağı veriler, işlemin nasıl yapılacağına belirten komutlarla birlikte verilir. İşlemci bellek içindeki komutların sırasıyla ger - çekleşmesini sağlar.

Bellek: Bellek, bit adı verilen çok küçük birimlerden oluşur. İşlemci çalışır haldeyken, bu birimler ya 0, ya da 1 konumunu gösterir. Bu konumlar birimler - deki manyetik alanların yönüne göre önceden belirlenmiştir.



Merkezi İşlem Ünitesi, yan elemanları tıpkı bir orkestra şefinin müzisyenleri kontrol ettiği gibi eder. Bilgiyi giriş terminalerinden, ya da belleklerden alır ve işlemler bittiğinde sonuçları yazıcıyla ya da görsel çıkış ünitesine gönderir. (Sağda)



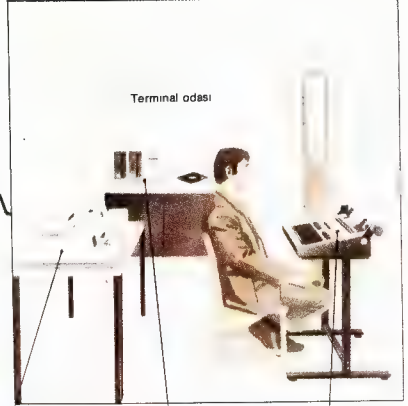
VDO Veriler, klavyeyle gönderilir, çıkışlar da ekranda belirir

CPU Bilgisayarın işlemeleri yapan ve yan üniteleri kontrol eden ünitesi

Kaset Okuyucu Veriler ve programlar ses bantları üzerine kaydedilir Çıkış, kağıtta yazılı olarak gelir

Disket Belleği Veriler ve programlar, ince manyetik diskler üzerine kaydedilir

Teleks Terminal. Veriler, klavyeyle gönderilir ve çıkış kağıda yazılır



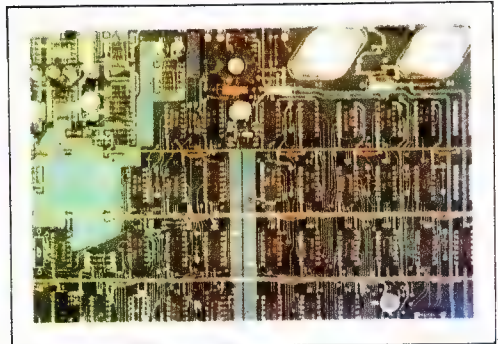
Ancak yeni tip bilgisayarlarda, bu işi çipler üstlenmiştir. Çip adı verilen özel entegre devrelerin içine yüzlerce **bitlik** hafıza saklanabilir. Yine önceden belirlenmiş bir sayıdaki **bitten** oluşan bir birime de **bayt** denir. Baytların her biri, bir standart karakter ya da sayı alabilecek kapasitededir.

Bir işlemcinin büyüklüğü ya da kapasitesi bir anda alabileceği ve belleğinde saklayabileceği bayt sayısına göre belirtilir. Kapasiteleri çok küçük, tek bir amaç için yapılmış, birkaç yüz **baytlık** belleği olan mikro işlemcilerinkinden, içinde milyonlarca baytlık bilgi saklayabilen büyük belleklere sahip bilgisayarlara kadar değişir.

Tüm yan ünitelerin aynı bölme içine sığdırıldığı, küçük bir bilgisayarın ana kartı. Her siyah dikdörtgen, özel bir çip taşımaktadır. Bunlardan biri, bilgisayarın işlenmesi. Diğerleri, ekrana gönderilen çıkışları kontrol ediyor ve programları ve verileri saklayan bellek görevini görüyor. (Alta sağda)

İşlemcinin, bilgi saklayabilme özelliğinden sonra, ikinci önemli özelliği, bellekten istenen bilgiyi bulup işleme koyma yeteneğidir. Bunu kolay bir hale getirebilmek için bellek, kendi içinde bölümlere ayrılır. Her bölümün yerini gösteren bir tanımlama vardır. Bu tanımlamalara **adres** de denir. Her bölümün büyüğü, bilgi-

sayardan bilgisayara değişir, bu bölümlere **kelime** adı verilir. Boyları, her bir **bayt**ın adreslenebileceği şekilde düzenlenmiştir. Bazıları ise, ki bunlar çok hızlı çalışan işlemcilerdendir, iki **bayttan** on iki **bayta** kadar değişen kelimeler kullanılır. Bunlardan birincisi, ikilik düzendir. İkincisi ise, ikilik düzence kodlanmış ondalıklardır.





Bilgisayar donanımlı baskı makineleri ureten bir fabrikadan görünüm. (Üstte)

İkilik düzen modunda, bir kelimenin içinde yer alan her bit sayısal bir değeri belirtmek için kullanılır. Bu formda, verilen bilginin sınırlandırılmış olması gerekmez. Bunun yerine, verilebilecek en uzun sayısal değerin sınırını doğrudan kelimenin uzunluğu, yani kelime içinde yer alan bitlerin sayısı belirler. Örneğin, kelimesi on iki bit uzunluğunda olan bir işlemde, 29 sayısı şu şekilde gösterilir.

$$29 = 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 11101$$

on iki bitlik kelimemiz:

00000011101

Bir işlemcinin kapasitesini belirleyen ikinci bir faktör de istenen bilgiyi bellekten doğru olarak bulma süresidir. Bu süreye erişim süresi denir. "Yavaş" bir işlemcinin erişim süresi milisaniyelerle ölçülür. Günümüzde erişim süresi 1 mikrosaniyeden kısadır. Hız, nanosaniyelerle (saniyenin mil - yarda biri) ölçülmektedir.

Yan Bellek: Bilgisayarda bir işlemin yapılabilmesi için, hem o işlemin yapılmasını sağlayan komutların, hem de kullanılacak verilerin bellekte olması gerekir.

Bu nedenle bellek, çoğu zaman bütün komutları ve verileri aynı anda saklamak için yetersiz kalır. Bu koşullarda, fazla veri ve komutları saklamak ya da arada yeniden kullanılabilecek bilgileri bir yere almak için yan bellek kullanılır. Bu belleklerden biri, **manyetik teyptir**. Bugün kullanılan bilgisayar sistemlerinin tümünde, **manyetik teyp, disk ve disket** gibi yan bellekler vardır. Bunlar, **Merkezi İşlem Ünitesi (CPU)** içinde yer alan belleğin çok daha fazlasını saklayabilecek kapasitedirler.

Çok uzun ya da çok fazla kullanımlı bilgilerde, **manyetik teypin** hızı ve kapasitesi yetersiz kalabilir. **Salt Okuyucu Bellek** (Read only Memory, ROM) adı verilen teyplerde, işlemcinin istenilen bilgiyi bulabilmesi için, teypi en başından itibaren okuması gerekir. Bu yüzden, teypten bilgiye ulaşmak için epey bir zaman harcar. Erişim zamanını kısaltmak için, daha gelişmiş bellek cihazları üretilmiştir. Bu cihazlarda, **bilginin adresi, bulunduğu yer, erişim süresini hemen hiç etkilemez**. Bu tip cihazlara **Rastlantısal Erişilebilir Bellek** (Random Access Memory, RAM) adı verilir. Bu cihazlarda da bilgiler, manyetik bir yüzeyin üzerine, aynen manyetik teypte olduğu gibi yazılır. Onlarda erişim zamanını kısaltan, aynı anda okuyup yazabilen bir-

den fazla kafa olmasdır.

Şimdi bu bellekleri biraz daha detaylı inceleyelim. **Manyetik teyp**, erişim zamanının çok uzun olması ve daha fazla yer kaplaması nedeniyle şu anda artık pek kullanılmamaktadır. Bu teypler, bizim evde kullandığımız teyplere çok benzerler. Üzerindeki bilgi de yine bizim teypte dinlememize benzer bir şekilde alınır. Teybin üzerindeki kafa, bir önce yazılmış olan bilgileri okur. Bu bilgiler, daha önce anlattığımız **modlardan** biri ile yazılmıştır. O'lar ve l'ler, yazım anında yaratılan manyetik alanların yönüne göre belirlenir.

Rastlantısal Erişilebilir Bellek (RAM) türleri, çeşitlidir. Yine bunlardan bir kısmı da pek kullanılmamaktadır. Biz artık kullanılmayan bellekler hakkında kısa bir bilgi verip geçeceğiz.

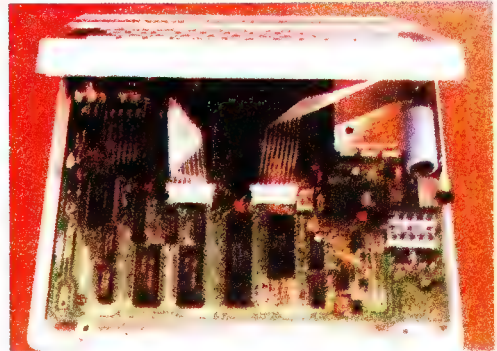
Bunlardan ilki, **kart okuyucudur**. Bilgiler, kartlarda saklanır. Daha önceden kart delicisi ile delinen kartlar, bu sistemde okunur. Kartlarda, her sayının karakterin ya da komutun bir yazılış yöntemi vardır. Daha önceden belirlenmiş, her sayı, karakter ya da komut için ayrı ayrı delinir. Bu şekilde delinen kartlar kart okuyucu tarafından okunur. Kart okuyucunun içinde, delinmiş ve delinmemiş bölgeleri hissedebilen kafalar vardır. Bu sayede saklanan bilgiler okunur.

İkinci bellek ünitesi, **manyetik silindirdir**. Silindir çalışır durumdayken sürekli dönmektedir. Bu arada çevresinde de okuyucu kafalar bulunmaktadır. Birden fazla kafa olduğu için, silindirin taranması fazla bir zaman almaz. İstenilen adres bulunduğu anda, ki bu ona en yakın kafa tarafından yapılır, bilgi hemen **Merkezi İşlem Ünitesine** aktarılır.

Disk ünitesi, günümüzde en çok kullanılan bellek ünitelerinden biridir. Diskin çalışması da silindrinkine benzer. Disk, üst-üste konmuş üç ya da daha fazla manyetik plakadan meydana gelir. Bu plakaların üzeri, manyetik bir tabakayla kaplanmıştır. Bu tabaka, aynen teypte olduğu gibi yazılan bilgileri korumak için konulmuştur. Bu plakaların iki yüzü de kullanılabilir. Ancak herhangi bir aksiliğe karşı en üst ve en alt yüzler kullanılmaz. Yani üç plakadan meydana gelen bir disk, alt yüzünden dördü kullanılır. Ancak yine bunlardan biri de diskin kullanımına ait bilgileri taşıdığı için, üç yüz üzerinde bilgi saklanabilir. Dördüncü yüz, diskin içindeki bilgilerin adreslerini saklar. Okuyucu kafalar, yazılan bu adreslerin yardımıyla istenilen bilginin yerini bulur. Adresler şu şekilde behlenir:

Disk'in yüzü aynen bir plağın yüzüne benzer. İççe geçmiş dairelerden oluşur. Bu dairelerin herbirine yol denir. Her yol, 0'dan başlayarak numaralanmıştır. Bilgiler, bu yollar üzerine yazılır. Yollar da, sektör denilen parçalara ayrılmıştır. Sektörler de numaralıdır. Bir bilginin adresini vermek için, önce yol, sonra da sektör numarası verilir. Bunu mahalle ve sokaklara benzetebilirsiniz. Nasıl ki biz, bir evi tarif ederken, önce mahallesini, sonra da sokağını söylüyorsak, burada da önce yolunu, sonra da sektörünü söylüyoruz.

Bir disk ünitesinin istenilen bilgiyi erimesi, ortalama 20 milisanیه alır. Bunları Merkezi İşlem Ünitesine aktarırken, çok daha hızlı çalışır. Sanıyede ortalama 90.000 karakterlik bilgiyi transfer edebilir. Bir diskin bilgi



Günümüzde her bilim dalında uygulanmaya başlayan bilgisayarlar sanıyede de söz sahibi oldular

saklama kapasitesi de bir milyara yakındır.

Son zamanlarda kullanılmaya başlanan bir bellek ünitesi de **diskettir**. Disketin yapısı diske çok benzer. Ancak diskten çok daha



Yönetim ve muhasebede kullanılan bir bilgisayar sistemi.

küçük, esnek ve pratiktir. Ortalama on beş santimetre çapında, minik bir plağa benzer. Kolay taşınabilir olması da kullanımını arttırmabilmektedir.

Bilgisayar bellek ünitelerine kısaca değindikten sonra, diğer ünitelere geçelim. Bunlar bilgisayar sistemiyle kullanıcı arasındaki iletişimi sağlayan ünitelerdir. Bilgi girişi ve çıkışı, bu üniteler aracılığıyla sağlanır.

Kart Delici: Kart okuyucuda olduğu gibi, **kart delici** ünitesi de artık kullanılmamaktadır. Kullanılan kartlar standarttır. Her kartta 80 kolon vardır. Bu 80 kolon, 12 sıra halinde altalta gösterilmiştir. Her kolon ve sıraya toplam 4096 değişik karakter yazılabilir. Örneğin K harfini yazmak için, kullanılacak kolonun ikinci ve on birinci sırası delinir. Delicinin önünde daktilo klavyesine benzer bir klavye vardır. Biz, karakterleri bu klavyeyle yazabiliriz. O, kendi içinde o karakterin delgi kodunu bulup kartı deler.

Manyetik teyp, disk ve disket de bilgi girişi için kullanılan ünitelerdir. Bunlardan başka, bilgisayarla doğrudan bağlantı kuran giriş terminaleri vardır. Bunlar,

iki çeşittir. Bu terminalerin de genelde, daktilonunkine benzeyen bir klavyeleri vardır. Çoğunlukla bu klavye doğrudan bilgisayara bağlıdır. Klavye ya da duyucu, makineye bir seri elektriksel uyarı gönderir. Sonra bu uyarılar, bilgisayarın anlayacağı ikilik düzene çevrilir.

Yazıcı: Bilgisayar çıkışlarının çoğu, yazılı olarak istenir. Bir yazıcıyla bu çıkışlar kâğıda dökülebilir. Yazıcılar, yaptıkları işe göre çeşitlere ayrılırlar.

Matris Yazıcı: Bu yazıcı, teleks makinesine benzer bir şekilde çalışır. Yazıcının üzerinde matris şeklinde noktaları bulunan bir kafa vardır. Karakterler, şekillerine göre belli kalıplar halinde yazılır. Bu kalıplar, önceden yazıcının hafızasına verilmiştir. Ona göre, Merkezi İşlem Ünitesinden gelen mesajlarla bu kalıplar karşılaştırılır ve istenen kalıp bulunur. Bu tip yazıcılarda, karakterler tek tek basılır. Bu şekilde harfler çıkmış olur. Matris yazıcılar en yavaş çalışan yazıcılardır. Saniyede on karaktere kadar basabilirler.

Satır Yazıcı: Satır yazıcılar da iki çeşittir. Bir çeşidinde, üzerinde karakterler basılı olan silindirdir kullanılır. Silindir sürekli kendi çevresinde döner. Bu arada

istenen karakter geldiğinde, silindir şeride o karakteri basar.

Diğer çeşidinde ise, üzerinde karakterlerin yer aldığı bir çesit zincir kullanılır. Bu zincir, şeridin hemen önünde sürekli dönmektedir. İstenilen karakter geldiğinde, zincir şeride bastırılır.

Bunlardan başka, bilgisayara giriş de yapabilen yazıcılar vardır. Bu yazıcıların, yukarıda belirtildiği gibi üzerinde klavyeleri bulunur. Bu klavyeler sayesinde, bilgi girişi ve çıkışı aynı üniteden yapılabilir.

Görsel Giriş Ünitesi: Bilgisayardan çıkan sonuçların saklanması gerekmiyorsa, **Görsel Çıkış Ünitesi** (Visual Display Unit, VDU) adı verilen televizyona benzer bir ünite kullanılır.

Bu ünite klavye ve buna benzer bir ekran vardır. Klavyede basılan her karakter, Merkezi İşlem Ünitesine giderken ekranda da görülür. Ayrıca, Merkezi İşlem Ünitesinden gelen mesajlarda da görüntüye girer. Bu şekilde, bilgisayarla doğrudan bağlantı kurulmuş olur.

Bütün bu yan üniteler, Merkezi İşlem Ünitesi tarafından kontrol edilir. Ne zaman belleklerden biriyle bilgi gönderileceğini, ne zaman giriş terminallerinden veri gerektiğini ve ne zaman hangi terminal ile çıkışı yapılacağını hep o düzenler.

Bahis Nedir?

“BELKİ”, “olası” sözcükleri bir şeyin olabileceğini, ama bunun kesin kes belli olmayacağını belirtir. Matematikte bir şeyin olma olasılığını hesaplayan bir yol vardır. Buna **olasılık** denir.

Olasılık: Bir şey mutlaka olacaksa, onun olasılığı 1'dir. Örneğin bir insanın herhangi bir gün öle olasılığı 1'dir. Hiçbir zaman olmayacak şeylerin olasılığı 0'dır. Bir para, birçok kez havaya

atılırsa, yazı ve tura gelme sayısı aşağı yukarı eşittir. Ortalama olarak, her iki atıştan birine yazı gelir. Yani yazı gelme olasılığı $1:2=0.5$ 'dir.

İnsanlar kumar oynarken, bir şeyin olma olasılığını hesaplamaya çalışırlar. Eğer bir turnuvada on altı takım yarışıyor, her birinin kazanma şansı eşittir. Eğer hepsi aynı şekilde iyi oynuyorsa, her birinin kazanma şansı $1/16$ dır. Ama genellikle bazı takımlar, diğerlerinden daha iyidir. Bahisçiler hesaplarını yaparken, bütün şartları düşünürler. Yarışın favorisine 1'e 3 verebilir-

ler. Bu, takımın, her yengiden sonra üç yenilgi alabileceği düşünülmüyor demektir. Ya da, ortalama her dört yarıştan birini kazanacağı anlamına gelir.

Rastlantısal Oyunlar: Çevremizdeki insanların işlerine nasıl gittiklerini öğrenmek istediğinizi varsayalım. Tabii her gördüğünüz insana sorabilirsiniz bunu. Ama bu binlerce insana soru sormak anlamına gelebilir. Dolayısıyla, diğerlerini de temsil edebileceğini umduğumuz birkaç insanla konuşur, ona göre karar veririz. Bu tip bir seçimle alınan



Rulet tam bir şans oyunudur. Topun, tekerin üzerindeki bir sayıya gelme olasılığı eşittir (Üstte)



Bir para atıldığında, tura gelmesi olasılığı tam 0.5'tir. Ortalama iki atıştan biri tura gelir. (Üstte),



Otobuse binme şansı: nedir? Otobusun dolu olduğu sıralar bu olasılık azalır. En az olasılık, kuyruğun arkasındakiler içindir (Üstte)



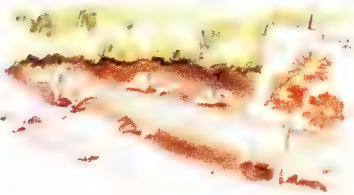
Bir piyangoda, her biletin kazanma şansının eşit olduğundan emin olmak gerekir. Bunun için biletler iyice karıştırılır (Üstte)

Londra yakınlarındaki Thames nehrine, taşkınları önlemek için bir engel yapmışlardı. Uzmanlar, taşkın olasılığının yılda 0.1 oranında olduğunu söylüyorlar. Bu, nehrin hiçbir zaman taşmayacağı anlamına gelmez, ama olasılık hesapları, engelin görevini yeterince yaptığını gösteriyor (Sağda)



örneklere rastlantısal örnekler denir.

Gerçek bir rastlantısal örnekleme, şaşılacak kadar zor olabilir. Yolda durup, geçen herkesle konuşmak, yeterince iyi olmaz. Çünkü bu şekilde, arabayla, ya da otobüsle gidenleri atlamış olursunuz. Onları da dikkate alsak bile,



Toprağına tohum eken bir çiftçi, bunların nepsinin yeşermeyeceğini bilir. Eğer yeşerme olasılığı azsa, fazladan tohum ekmesi gerekir. (Üstte)



Kâğıt oyunlarında, oyuncular en iyi grup kâğıdı toplamaya çalışırlar. Doğru kâğıdın gelme olasılığın, iyi hesaplayan oyuncuların kazanma olasılığı daha fazladır. (Üstte)



Bahisçiler, bahis hesaplarını yapmadan önce her atın kazanma olasılığını tahmin etmeye çalışırlar. Kazanma olasılığı yüksek olan atların bahisleri küçüktür. (Üstte)



Her zar atışta, atanın istediği sayının gelme olasılığı 1/6'dır. Birden fazla zar kullanılırsa, olasılık artar. (Üstte)



Günümüzün en yaygın bilgisayarları, elde taşınabilen ve standart kasetlere oldukça fazla bilgi alabilen, orta boy hesap makinesi büyüklüğündeki aygıtlardır.

trenle gidenleri atlarız. Bu yüzden, örnekleme yaparken çok dikkatli olmamız gerekir.

İstatistikler: Rastlantısal örneklerden ya da futbol maçlarının sonuçları gibi konulardan çıkan sonuçlardan, sayısal bilgiler çıkarılabilir. Bu bilgiler toplandığında, söz konusu olay hakkın-

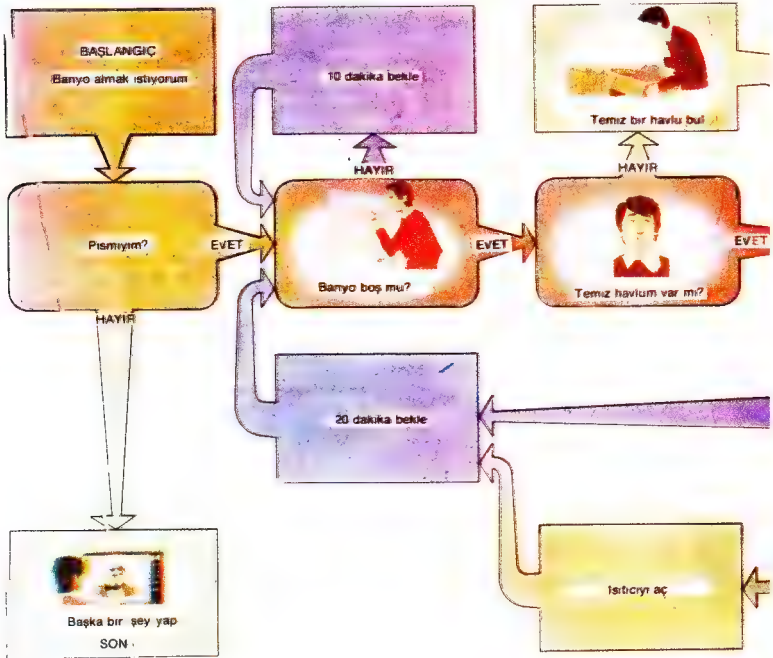
da bir istatistik elde edilmiş olur.

İstatistiklerin büyük bir dikkatle hazırlanması şarttır. Örneğin, kaza istatistikleri, çoğu kazanın sürücünün evine yakın yerlerde meydana geldiğini gösterir. Bu, sürücülerin kendi bölgelerinde daha dikkatsiz oldukları anlamına gelebilir. Çoğu gezintilerin kısa olduğu, bu yüzden sürücülerin evlerine yakın olduğu anlamına da gelebilir. Gerçek nedeni bulmak için, sürücüler hakkında daha geniş bir istatistik yapmak gerekir.

Bilgisayarlardaki bu gelişmeler bir bilimi dalı olarak ele alınan ve uygulamaları istatistiğin de gelişimine katkıda bulunmak tadır.

Bu nedenle de yukarıda verilen örnekte olduğu gibi yalnızca kazalarla ilgili istatistikleri değil, bir insanın ne zaman hastalacığıyla ilgili verileri de elde etmek mümkündür.

Buradan yola çıkılarak da bir insanın tüm sağlığıyla ilgili bilgiler yığıldığında elbette ne zaman iyileşeceği birer istatistik sonucu olarak saptanabilir.



Bilgisayar da Her Şey Adım Adım

MANTIK, problemleri adım adım, çözmenin bir yoludur. Çok karmaşık problemler bile, mantık yoluyla daha basite bölünerek çözülebilir. Bunun için her adım sonucunu bilmek yeterlidir.

Oyunlar ve Mantık: "Kızma Birader" gibi oyunlar, bütünüyle şansa bağlı oyunlardır. Mantık hesaplarının sonuca pek yararı olmaz. Her şey zara göre değişir. Ama, "Dama", "Satranç" gibi

oyunlarda da şansın hiç yeri yoktur. Oyuncular, belli kurallara göre taşlarını hareket ettirirler. Oyunun sonucu, sadece ve sadece bu hareket ettirilen taşlarla belirlenir.

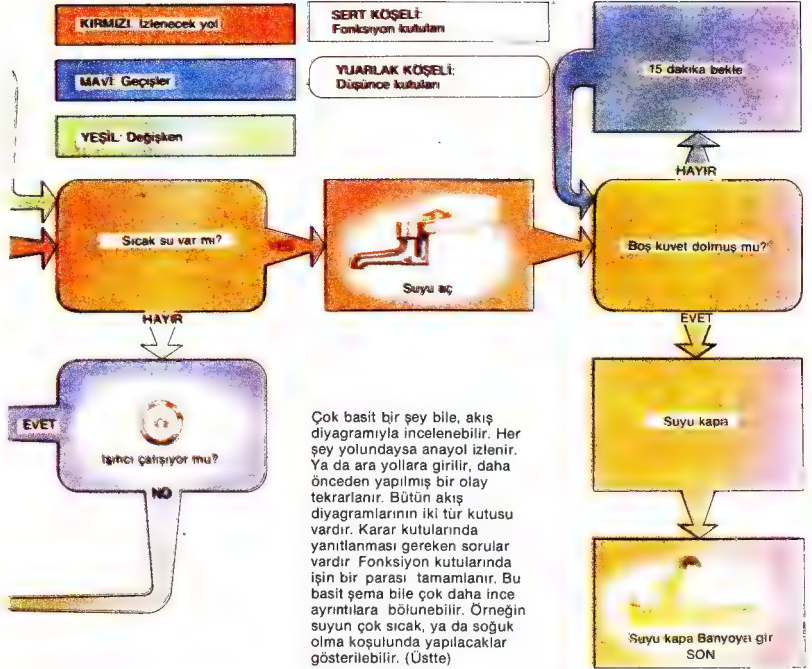
İyi bir oyuncu şöyle düşünür: Eğer ben, şunu oynarsam karşı taraf da bunu ya da şunu oynar. Bu benim durumumu biraz daha düzeltir ve.. Bu şekilde gider. Bir oyuncu ne kadar ilersini düşünebilirse o kadar iyi oynar.

Bilgisayarlar ve Mantık: Bilgisayar mantıkla çalışır. Problemleri çözmek için önce onları çok basit adımlara indirger. Sonra da yanıt bulana dek adım adım ilerler.

Bilgisayarlar, mantık oyun-

larında çok başarılıdır. Programın içine kaydedildiği için hiçbir kuralı unutmazlar. Bir bilgisayar oyun ilerledikçe öğrenmek üzere programlanabilir. Bunu karşı tarafta oynayan insanın yaptıklarını adım adım not ederek başarır. Kendi hamlesini yapmadan önce, daha önce buna benzer bir durumun olup olmadığını araştırır. Eğer varsa, rakibinin buna karşı ne yaptığını bulur, böylece rakibinin taktığını öğrenir. Bu şekilde programlanmış bir bilgisayarı yenmek çok zordur.

Akış Diyagramları: Bütün karmaşık problemler ve işlemler basit parçalara ayrılabilir. Bir problemin bu şekilde basitleştiril-



Çok basit bir şey bile, akış diyagramıyla incelenebilir. Her şey yolundaysa anayol izlenir. Ya da ara yollara girilir, daha önceden yapılmış bir olay tekrarlanır. Bütün akış diyagramlarının iki tür kutusu vardır. Karar kutularında yanıtlanması gereken sorular vardır Fonksiyon kutularında işin bir parçası tamamlanır. Bu basit şema bile çok daha ince ayrıntılara bölünebilir. Örneğin suyun çok sıcak, ya da soğuk olma koşulunda yapılacaklar gösterilebilir. (Üstte)

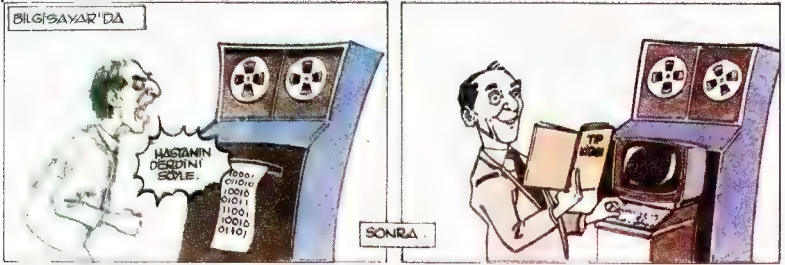
miş halini gösteren diyagramlara akış diyagramı adı verilir. Akış diyagramı bizi bir dizi kutular ve oklarla adım adım sonuca yaklaştırır. Herhangi bir soru sorulduğunda bunun kesinlikle evet ya da hayır yanıtı olması gerekir. Değişik yanıtlar bizi diyagramın değişik yerlerine götürür. Akış diyagramları her türlü yolu ya da problemi gösterebilir. Matematikte bir denklemin nasıl çözüleceğini, banyoya girerken ya da bir gömlek alırken neler yapmamız gerektiğini belirtirler.

Doktorlar hastalarının sorunlarını bilgisayarın yardımıyla öğrenir. Hasta, doktor yerine bilgisayarla iletişim kurar. Bilgi-

sayar sorularını ekrana çıkarır, hasta da evet ya da hayır diye cevaplar verir. Bilgisayar sorularını hastanın verdiği yanıtlara göre seçer. Örneğin, "Eğer bilgisayar akşamları iyi uyuyabiliyor musunuz?" diye sorarsa, hasta "hayır" diye yanıtlarsa, bu durumda uyku ile ilgili daha fazla sorular soracaktır. Eğer hasta "evet" derse başka bir konuya geçecektir.

Doğaldır ki, kimse bu bilgisayarların doktorların yerini almasını beklemiyor. İyi bir doktorun anlayışlılığını hiçbir bilgisayar gösteremez. Ama basit ve zaman alan işlerde bilgisayarlar, doktorların en büyük yardımcılarıdır.

İşte bir mantık oyunu daha. Amaç x ya da 0'ı yan yana getirebilmektir. Mantıksal yol, her x'in, ya da her 0'ın yanına bir diğerini koymaktır. (Üstte)



Bilgisayar Kullanımı ve Programlanması

BİR bilgisayar düzeni, çarısı elektronik parçalar dan oluşur. Kendi kendine düşünemez. Ne yapması gerektiği, en ince ayrıntısına kadar söylenmelidir ki, düşünmeden sonucu çıkarabilsin. Bir düzen ne kadar büyük olursa olsun, ne kadar hızlı çalışırsa çalışsın, ona gereken komutlar verilmediği sürece hiçbir işe yaramaz.

Programlama ve Kullanım: Bir



bilgisayara neyi nasıl yapacağını gösterme işine **programlama** denir. Programlamada kullanılan komut dizisine de **program** denir. Bilgisayarda kullanılan programlara, **bilgisayar kullanımı** adı verilmiştir. Düzenin kendisi ile yapılan çalışmalar da bilgisayar donanımı ile ilgilidir.

İki tür bilgisayar kullanımı vardır. Her bilgisayar, kendini ve kendine bağlı yardımcı ünitelerini nasıl çalıştıracağını gösteren bir komut dizisine gereksinim duyar. Bu, düzen kullanımıdır. Bu olmadan, makine hiçbir iş yapamaz. Genellikle bilgisayar firmaları, bu komut dizisini kendileri hazırlayıp (program haline getirip), bilgisayarın belleğine

silinmeyecek bir şekilde yerleştirirler.

Özel bir işi yapmak için verilen programlar da, uygulamanın kullanımı içine girer.

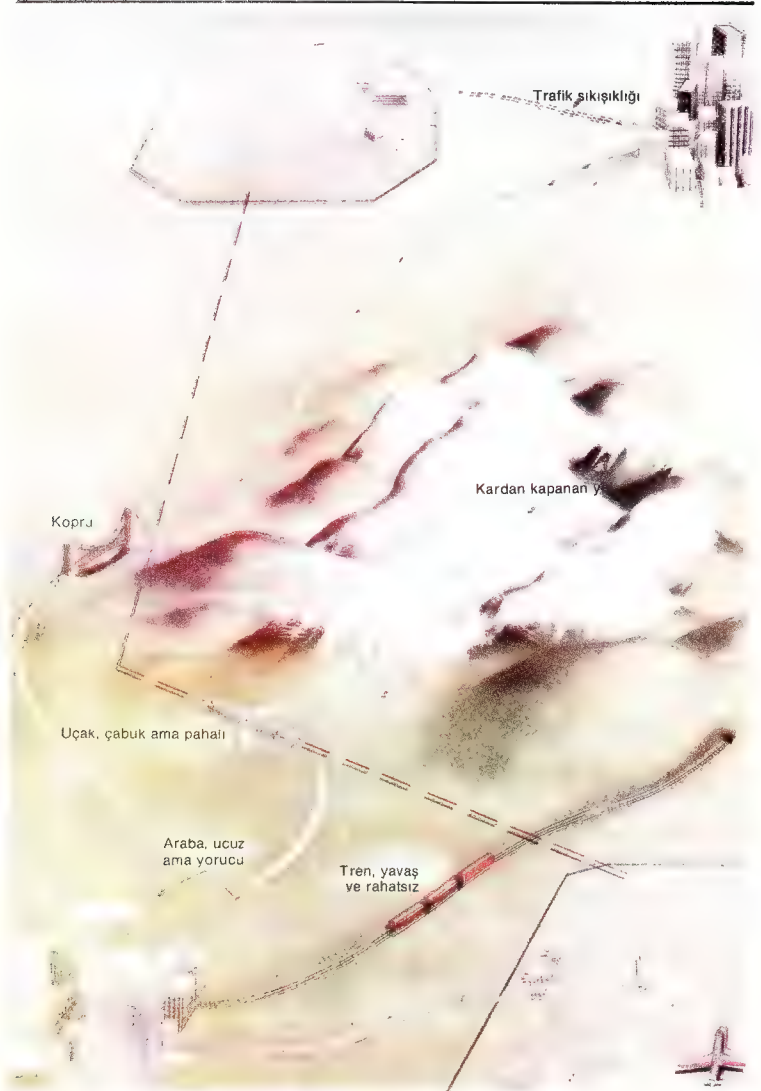
Program Dilleri: Bilgisayar, ikilik düzene göre çalışır. Bu yüzden programların da ikilik düzende olması gerekir. Bir programı 1'ler ve 0'larla yazmak çok uzun ve yorucu olur. Özellikle binlerce komut taşıyan programları yazmak günler alır. Bu nedenle, programcıya yardımcı olmak amacıyla çeşitli diller yaratılmıştır.

Kullanıcı, programları bilgisayara bu dillerden biriyle verir. Sonra bilgisayar onları ikilik düzene kendi içinde çevirir. Bu çevirme işi de, bilgisayarın düzen kullanımına girer.

En çok kullanılan dillerden biri, **BASIC**'tir (Beyzik okunur). Her tür program **BASIC** dilinde yazılabilir. Ayrıca küçük kapasiteli bilgisayarların tümü, **BASIC** dilini kullanır. Bu dil diğer bütün diller gibi yabancı dil öğrenmekten çok daha kolaydır. Çünkü öğrenilecek çok az kelimesi vardır.

Ayıklama: Özel bir iş için bir program yazıldığında, ikinci adım onu kontrol etmektir. Bir programın yazıldığı anda çalışması, çok az rastlanan bir olaydır. Genelde, mutlaka programın bir yerinde bir hata vardır. Bu hatayı bulup düzeltme işine **ayıklama** adı verilir.

Bazen bir program, bir süre için çalışabilir. Programcının koymayı unuttuğu bir komut yüzünden de birden durabilir.



Modern bir bilgisayarın klavyesi, kullanıcının program ve verileri doğrudan makineye ulaştırmasını sağlar. Genellikle yazılanları anında gösteren bir VDU ekranına bağlanmıştır. Bu şekilde, kullanıcı hatalarını görebilir ve hemen düzeltebilir. Kursör denilen, yanıp sönen küçük bir nokta, bir sonraki karakterin nereye basılacağını gösterir. Düzeltme yapılacağı zaman kursör, yanlış yazılan yerin üzerine getirilir. (Sağda).

Alfabetik tuşlar, daktilonunkine benzer bir düzenle sıralanmıştır. Genellikle ilk altı karakteri sırasıyla isimlendirilmiştir.

Kullanıcı, bu tuşları kursörü hareket ettirmek için kullanır. Kursör, her yöne hareket ettirilebilir.

Fonksiyon tuşları, satır başına dönüş, hata düzeltme, sayfa atlama gibi özel amaçlar için kullanılır.

Nümerik tuşlar bir araya getirilmiştir. Rakamların düzeni, hesap makinelerinde kullanılanın aynısıdır.



Bilgisayarlarla İletişim Kurma

BİLGİSAYARLARDA, bilgisayar kullanıcıları bir bilgisayar dilinden anlarlar. Ancak bilgisayarların gözü ve kulağı yoktur. Özel bir araca

bağlanmadığı takdirde hiçbir bilgisayar kâğıda yazılmış ya da başka bir şeye kaydedilmiş bir bilgiyi okuyamazlar. Bilgisayara bilgi girişi, ancak onun anlayabileceği şekilde yapılır. Bunun için de özel birtakım yollar saptanmıştır.

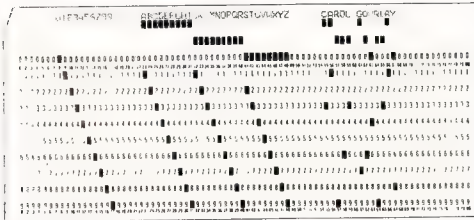
Delgi Kartları: Bilgisayara

bilgi girişi yapmanın en ucuz ve basit yöntemi, en eski olanıdır. Onsekizinci yüzyılın başından beri bilgisayarlarda kart delme işlemi kullanılmaktadır.

Kart delme ünitesini açıklarken, delgi kartları konusuna da değinmiştük. Bu kartlarda, her karakter için iki ya da üç delik açılır. Her kartın 80 ya da 90 karakterlik kapasitesi vardır ki, bu en basit komutları içine sığdırabilir. Bir bütün program, her kartta bir komut olmak üzere bir tomar kâğıda sığdırılabilir.

Delgi kartlarını kullanmanın bir avantajı, herhangi bir hata çıktığında ya da değiştirilmek istendiğinde yapılacak tek şeyin kartı değiştirmek olmasıdır. İstenmeyen kartlar çıkartıldığında, yeni programımız elimizde hazır olur.

Direk Giriş: Modern bilgisayar düzenlerinde, veriler bilgisayara doğrudan bağlanan bir klavyeyle ulaştırılır. Bu şekilde ulaştırılan bilgilere direk giriş denir. Klavyede, aynen daktiloda olduğu gibi harfler vardır. Ayrıca rakamlar için ve bilgisayara giriş yapıldığında ne yapması gerektiğini gösteren özel fonksiyonlar için de ayrı ayrı tuşlar vardır.



Bir delgi kartı (gerçek boyutlarının üçte ikisi boyutunda gösterilmiştir.) Her harf ya da rakam, bir ya da iki delikli bir kodla gösterilmiştir. Örneğin '4' rakamı, 4'üncü satırın delinmesiyle 'U' harfi de 0'inci ve 4'üncü satırların delinmesiyle gösterilir. Delgi makinesiyle bu delikler, kullanıcık klavyeye bastığı anda tam

yerlerine konulur. Aynı zamanda da ona karşılık gelen sembolü, kartın tepesine basar. Kartların delinmesi bittiğinde, okuyucuya verilir. Okuyucu, hissettiki her deliği, bilgisayara elektronik uyarılar halinde gönderir. Okuyucu, saniyede birçok kâğıt okuyabilir. Bu şekilde, çok uzun veriler bile çok kısa bir zamanda bilgisayara ulaştırılır.



Terminal, genelde bir VDU ekranıyla birlikte çalışır. Veriler, tuşlara basıldığı anda ekranda da belirir. Bir bilgisayar, verilerin kontrolünü yapmak için de programlanabilir. Örneğin adreslerin posta kutusunun yazılıp yazılmadığını kontrol edebilir. Eğer bir hata görürse, kullanıcıya ekran aracılığıyla bildirir.

OCR ve Çizgi Kodlama Düzeni: Bir bilgisayar, özel bir takım işaretleri anlamak üzere programlanabilir. Buna optik karakter teşhisi (optical character recognition - (OCR-) denir. Bu şekilde, kullanıcı oraya bir anahatar koymadan, verileri doğrudan sokabilir.

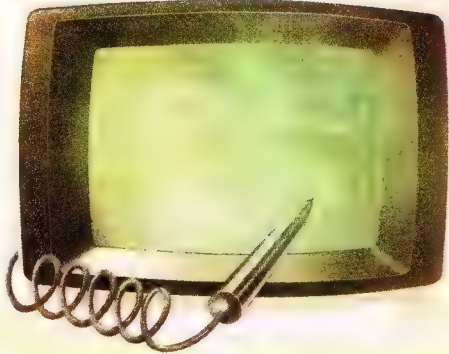
Kullanmakta olduğumuz bazı eşyalarda, birtakım siyah-beyaz çizgiler dikkatimizi çekmiştir. Bunlar çizgi kodunu meydana getirir. Bu çizgilerin her bir dizilişi, ayrı bir kod numarasını karşılar. Her maddenin ayrı bir kod numarası vardır. Bazı mağazalarda, bu kodları teşhis edebilecek bir alet konmuştur. Bu alet sayesinde, kasiyer toplam hesabı çıkarırken her malın tutarını ezberden bilmek zorunda kalmaz.

Bu alet sayesinde, bilgi yeniden bilgisayara gönderilir. Bilgisayarda malın cinsini ve tutarını

ekranda gösterir. Her giriş sırasında tutarları birbirine ekler. Aynı zamanda hangi malın ne kadar satıldığını da hesabını tutar. Bu şekilde, satış elemanı, hangi malın daha çok sattığını, bilgisayardan rahatlıkla öğrenebilir. Ayrıca bilgisayar, otomatik olarak, biten bir malın siparişini verebilir. Böylece, bir malın ansızın bitmesi diye bir sorun olmaz.

Bu süpermarketin çıkışında kasa bulunmuyor. Bilgisayar, malların üzerindeki çizgi kodlarını okuyup, ona göre toplamı çıkartıyor. (Ustte).

Kullanıcı, doğrudan VDU ekranının üzerine 'çizebilir'. Kalem ucunu hisseden programlar, o noktayı parlatır (Altta)





Tekbir diskin
üzerine 2500 iş
mektubu alınabilir



Ortalama 50 sayfalık
bir kitap, diskin
altmışta birini kaplar



On tane 10 000
kelimelik roman, bir
diske sığabilir

BİLGİ SAKLAMA

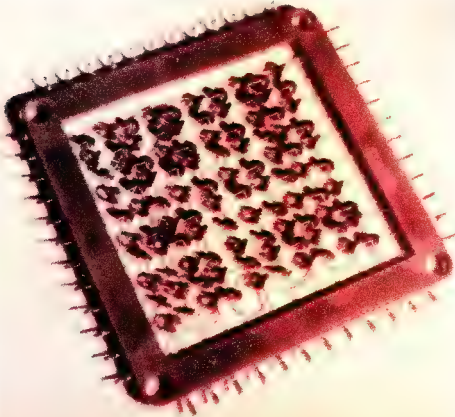
BİLGİSAYARIN kullandığı bilginin, üç yerde saklanması gerekir. Bunlardan biri, merkezi işlem ünitesi-

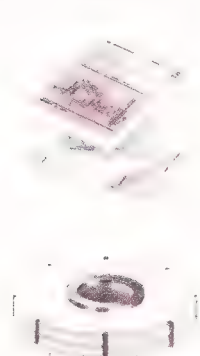
dir. Ayrıca merkezi İşlem Ünitesi(CPU)'nin dışında, ancak kolayca çıkabilecek bir yerde de saklanması gerekir. Bu ikisine ek olarak bir de her zaman el altında olmayan bir yerde de saklanması gerekir.

ROM ve RAM: Bilgisayarın iç belleği, kâğıdının üzerinde çalıştığı bilgileri saklar. Bu belleğin çok büyük olması gerekmez. Ancak belleğe gidip gelen bilgilerin, bilgisayarın hızını aksatmaması gerekir. Bunun için de, belleğin çalışma hızı oldukça fazla olmalıdır. Günümüz bilgisayarlarında bu işi, minik elektrik devrelerinden meydana gelen **çipler** görüyor. Eğer belleğin kapasitesini arttırmak istersek, yeni çipler ekliyoruz.

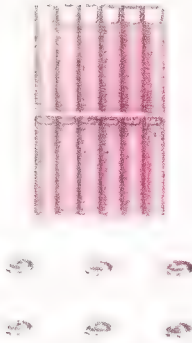
İç bellek iki kısma ayrılır. Bunlardan biri ROM (Salt Okunabilir Bellek), diğeri de RAM (Rastlantısal Erişilebilir Bellek) dir. ROM yani Salt Okunabilir Bellek, bilgisayarın çalışması için gereken en önemli bilgileri içinde tutar. Eğer bu programlar kaybolursa, makine çalışmaz. ROM'dan hiçbir bilgi çıkarılmadığı gibi, hiçbir yeni bilgi de eklenemez. RAM-yani Rastlantısal Eri-

Bir manyetik birim bellek ünitesi ilk bilgisayarlar, rastlantısal erişilebilir belleklerini bu ünitelerden meydana getirirlerdi. Her halka, tellerden akın geçirilerek iki değişik şekilde manyetize edilebilir. Çipler de aynı yöntemle çalışırlar. Ama çok daha küçüktürler. (Solda)

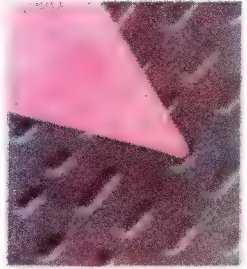




İstanbul telefon rehberi, üç diske sığdırılabilir.



Bir ansiklopedi seti, 30 diske, ya da altı disk ünitesine yazılabilir.



Bir disk yüzünün 1000 kez büyütülmüş şekli. Bilgi, diskin üzerine küçük delikler halinde yazılır. Lazer ışınları yardımıyla bu yazılar okunur. (Üstte)

şilebilir Bellek, tekrarlanarak kullanılabilir. Örneğin, disk, teyp gibi bellekleri tekrar tekrar kullanılabiliriz.

Dış Bellekler: Bilgisayarın dışında bilgi saklamak için en çok kullanılan üniteler, teyp ve diskettir. Özel bir kafa teyp üzerinden geçtikçe bilgiyi üzerine aktarır gerektiğinde de okur.

Manyetik diskler, manyetik oksitle kaplanmışlardır. Bilgi, üzerlerine tek merkezli daireler içine yazılır. Disk dönmeye başlayınca, okuma/yazma kafaları çok hızlı hareket eder. Bu şekilde, diskin herhangi bir yerine çok kısa bir sürede ulaşabilirler.

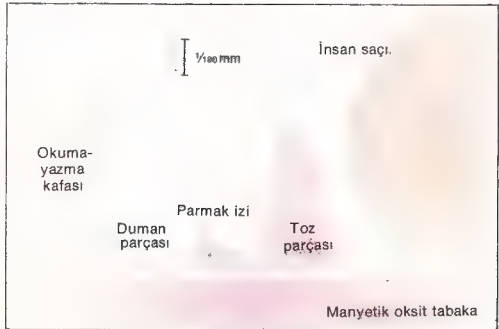
Bilgisayarlar, ayrıca disket de kullanırlar. Disketin çapı 12-20 cm. arasında değişebilir. Disketler, kendi boyutlarında bir koruyucu zarf içinde saklanırlar. Zarflerin üzerinde bulunan bir delikten okuma/yazma işleri yapılabilir.

Manyetik diskin okuma yazma kafalarını gösteren bir şema. Kafalar ve disk arasındaki uzaklık o kadar küçüktür ki çok küçük bir toz parçası bile, yüzeyin çizilmesine neden olur. Bunu önlemek için, disklerin çok temiz bir atmosferde saklanması gerekir (Sağda)

Disketler genelde beş-altı disk plakasının üst üste konmasından meydana gelirler. Kafalar, hiçbir zaman bu plakalara deymez. Aralarında çok duyarlı bir açıklık vardır. Bu açıklık ne kafaların işlerini aksatacak kadar fazla, ne de kafaları plakalara değdirecek kadar az olmalıdır. Bu aralık, santimetrenin yüzde biri uzunluklarla ölçülür. Kafalar, plakalara değdiği an hem kafa, hem plaka çizilir ve bir daha kullanılmayacak hale gelir. O yüzden bu aralığın bozulmamasına çok dikkat etmek gerekir.

Manyetik diskler, büyük çapta bilgiyi saklayabilecek kapasitededirlir. Bir disk plakası, bir milyon ya da daha fazla karakteri saklayabilir. Teknik geliştikçe, daha küçük disklere daha çok bilgi saklanabiliyor. (Solda)

Yedek Bellek: Asıl bellekteki bilgi genellikle başka bir yere daha kopya edilir. Bu yedek kopyalar, güvenli, yangın tehlikesinden uzak bir yerde korunurlar. Bu kopyalama işi herhangi bir olay sırasında eldeki bütün bilgilerin kaybolmaması için yapılır.



Bilgi Çıkışı

BİLGİSAYAR, işlemlerini bitirdikten sonra, sonucu dışarıya bir şekilde aktarması gerekir. Bilgileri alan kişi bir kullanıcı, bir şirketin satış servisi, bir müşteri, ya da bir işletme olabilir. Bilgi çıkışının düzeni, gerekli olan işe uygun bir şekilde ayarlanabilir.

VDU Ekranları: Bilgi akışını yapmanın en kolay ve en ucuz yöntemlerinden biri, VDU ekranı kullanmaktır. Bilgilerin çağrılıp ekrana çıkması, hemen hemen aynı zamanda olur. Özellikle, çok kısa aralıklarla istenen ama saklanması da gerekmeyen bilgiler, bu yoldan elde edilir. Bazı küçük bilgisayarlar, özel ekran bile kullanmazlar. Bunları normal televizyona bağlamak gereklidir. Ekrandaki karakterler, çok küçük noktalardan meydana gelir. Genelde beyaz zemin üzerine siyah, ya da siyah zemin üzerine beyaz veya yeşil karakterler kullanılır. Bilgisayarlar, renkli ekranlara tamamen renkli çıkış yapabilecek şekilde de programlanabilir. Bu yöntem özellikle grafik ve şema çıkarımında kullanılır. Bazı VDU'ların içine küçük bir işlemci yerleştirilmiştir. Bunlar bazı işlemleri ana işlemciye göndermeden hemen çıkarabilirler. Bu tip VDU'lara zeki terminaller adı verilir.

Yazılı Çıkış: Birçok nedenden ötürü kullanıcı, çıkışın saklayabileceği bir şekilde olmasını ister. Bu, sonradan kullanılmak ya da başkalarına gönderilmek amacıyla olabilir. Bunun için bilgisayarlar yazılı çıkış verirler.

Yazıcılar, içlerinde bulunan herhagi bir çeşit kalıpa, mürekkepli şeridin üzerine vururlar. Bu şekilde, normal daktiloda olduğu gibi şeridin arkasında bulunan kâğıda karakterler çıkmış olur.

Bazılarında, elektrikli daktiloda olduğu gibi bir top kullanılır. Karakterler bu top üzerine basılmışlardır. Bu yazıcılardan verilen çıkışlar çok temiz olmakla

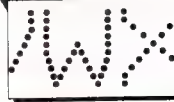


Bilgisayar tarafından çıkarılan bir hesap kâğıdı. Banka, bilgisayarla, açılan her hesabı yazıp saklaması için programlayabilir. Sonradan istendiğinde, müşterinin adresi dahil olmak üzere istenen her bilgiyi verebilir. (Altta)

Ufak bir bilgisayarla, bellekte bulunan bilgiler renkli televizyon ekranından çıkarılabilir. Yukarıda, ekrana bilgisayarın belleğine daha önce verilmiş listeler çıkıyor. (Üstte)

VADESİZ HESAP KARTI No 32 589 410		Sen No 349 19 EYL 80
Tarih	Açıklama	Tutarı
29 AĞS C	000387 Çek	50,00
1 EYL T	506000	20 39
4 EYL C	00389 Çek	50 00
5 EYL C	000384	9 25
8 EYL C	00388	1 35
15 EYL C	000390	128 59
	Faiz	363 25
BN S.S. GÖKAY SULUN CAD NO.10 LEVENT İST		
Önceki toplam	20 AĞUS 80	570 85
Toplam çek		259 58
Toplam faiz		363 23
Şimdiki toplam	19 EYL 80	674 50

○	US ASCII	○
○	"###%'0++,- /0123456789,(<=>?@ABCDEFGHIJKLMNopqrstuvwxyz\`" _'abcdefghijklmnopqrstuvwxyz()"	○
○	Swedish I	○
○	"###%'0++,- /0123456789,(<=>?@ABCDEFGHIJKLMNopqrstuvwxyzáâä"	○
○	Swedish II/Finnish	○
○	"###%'0++,- /0123456789,(<=>?@ABCDEFGHIJKLMNopqrstuvwxyzäöåü	○
○	Danish/Norwegian:	○
○	"###%'0++,- /0123456789,(<=>?@ABCDEFGHIJKLMNopqrstuvwxyzæ&öü _'abcdefghijklmnopqrstuvwxyzæ&öü	○
○	German	○
○	"###%'0++,- /0123456789,(<=>?@ABCDEFGHIJKLMNopqrstuvwxyzäöü _'abcdefghijklmnopqrstuvwxyzäöüß	○
○	British.	○
○	"###%'0++,- /0123456789,(<=>?@ABCDEFGHIJKLMNopqrstuvwxyz\`" _'abcdefghijklmnopqrstuvwxyz()"	○
○	Dutch	○
○	"###%'0++,- /0123456789,(<=>?@ABCDEFGHIJKLMNopqrstuvwxyz\`" _'abcdefghijklmnopqrstuvwxyz()"	○
○	Italian	○
○	"###%'0++,- /0123456789,(<=>?@ABCDEFGHIJKLMNopqrstuvwxyz*çé" _'abcdefghijklmnopqrstuvwxyzâêî	○
○	French/Belgian	○
○	"###%'0++,- /0123456789,(<=>?@ABCDEFGHIJKLMNopqrstuvwxyz*çé" _'abcdefghijklmnopqrstuvwxyzéûê	○
○	Spanish:	○
○	"###%'0++,- /0123456789,(<=>?@ABCDEFGHIJKLMNopqrstuvwxyz*ñü" _'abcdefghijklmnopqrstuvwxyz*ñü	○
○	Japanese	○
○	"###%'0++,- /0123456789,(<=>?@ABCDEFGHIJKLMNopqrstuvwxyz\`" _'abcdefghijklmnopqrstuvwxyz()"	○
○		○
○		○
○		○
○		○
○		○



Matris yazıcının çıkış kâğıdı. Yazıcı, çok değişik karakterlerle yazı yazmak, hatta grafik çizmek için bile programlanabilir. Ama karakterler biraz bulanıktır ve eğer büyütülürse noktalardan meydana geldiği görülür. (Üstte)

birlikte, fazla hızlı verilmezler. Matris yazıcılar, karakterleri, bunlardan daha az net çıkardıkları halde, hem daha çabuk çalışırlar, hem de grafik çıkarabilir, basit resimler yapabilirler.

Bu yazıcıların en büyük dezavantajı, fazla gürültülü olmalarıdır. Termal yazıcılar daha sessiz çalışır. Bu tip yazıcılar, karakterleri ısıya duyarlı bir kâğıta basarlar. Elektrografik ve elektrostatik yazıcılar kendilerine özgü kâğıtlar kullanırlar. Elektro grafik yazıcıların kâğıtları, ısıya duyarlı kâğıtlardır.

Jet-mürekkep yazıcıları, kâğıda ufak mürekkep damlaları göndererek çalışırlar. Mürekkep, elektronik olarak karakterlerin şekillerine göre yönlendirilir. Bu tip yazıcılar, çok değişik karakterlerde yazılar yazabilirler. Dahası kuramsal olarak imza atma yetenekleri de vardır. Ama bu, şimdiye kadar yaşamda gerçekleştirilememiş bir olgudur.



Yandaki toplar, yazıcılarda net karakterler elde etmek için kullanılır. Çok temiz yazırlar, ama karakter kapasiteleri azdır.

Karakterler, ya topların üzerinde kabartma şeklindedir ya da topun ucuna yerleştirilmiştir. (Üstte)

Görsel Bilgi

PEK çok bilgi, rakamlarla ilgilidir. Bunların bir bakışta anlaşılabilmesi için görsel ya da resimli bir hale çevrilir. Bu, bilgiyi daha kolay anlaşılabilir duruma sokar.

Şemalar: Bir restoranda 30 kişinin yemek yediğini düşünelim. Ismarlanan yemekler de şu şekilde olsun.

Kızarmış tavuk	: 10 kişi
Balık	: 8 kişi
Spagetti	: 7 kişi
Hamburger	: 4 kişi
Biftek	: 1 kişi

Bu yuvarlak bir şemada gösterilebilir. Şemada toplam, tam bir daireyle gösterilir. Daire, gösterileceklerin tutarıyla orantılı bir şekilde parçalara ayrılır. Parçalardan her birine **dilim** denir. Bizim elimizde toplam 30 kişi olduğu için, önce biz daireyi 30 eşit parçaya bölelim. Dairenin 360 olduğundan yola çıkarsak, her dilimin $360:30=12$ 'lik bir parçadan oluşması gerekiyor.

Her dilim, bir tabak yemeği gösteriyor. Bir kişi biftek yediğine göre biftek dilimi 12 olacak. Dört kişi de hamburger yediği için, hamburger dilimi de 4×12 48'lik oluyor.

KİM NE YEDİ?

Resim



10 tavuk

8 balık

7 spagetti

4 hamburger

1 biftek

Grafikler, şemalar, resimler vb. bize bilgiyi görsel olarak sunar. Her metodun kendine göre bazı avantajları ve dezavantajları vardır. Resimler çok basitür ve bize bilgiyi doğrudan aktarırlar. Ama buna karşılık detaylı bilgi veremezler. Grafikler, oldukça kesin sonuçlar verir,

ancak, onlar da pek çarpıcı değildirler.

Grafikler ve cetveller dikkate incelenmediği zaman insanı çok yanlış yollara sürükleyebilirler. Bazı grafikler, kendi içinden yanlışlıklar. Örneğin değerinden çok fazla gösterebilirler.



boyunca ne kadar araba geldiğini gösterecek olursak:

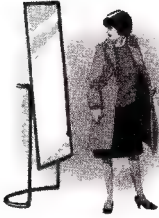
Pazartesi	:570
Salı	:520
Çarşamba	:530
Perşembe	:510
Cuma	:620
Cumartesi	:320
Pazar	:150

Her arabayı bir resimle göstermeyiz. Bu yüzden 100 arabayı bir resimle göstereceğiz.



Resimler: Aynı bilgi, birden fazla yolla gösterilebilir. Müşterilerin istediği yemekler, her tabak için değişik bir sembole, daha basit bir biçimde sunulabilir. Her sembol, temsil ettiği yemeğin şeklinde olursa, bir bakışta kimin ne yediği çok açık bir şekilde anlaşılır.

Resimlerle çok daha büyük sayılar da gösterilebilirler. Bir araba parkını ele alalım. Bir hafta

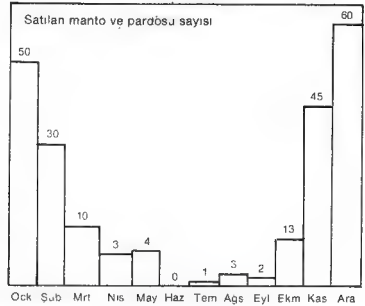
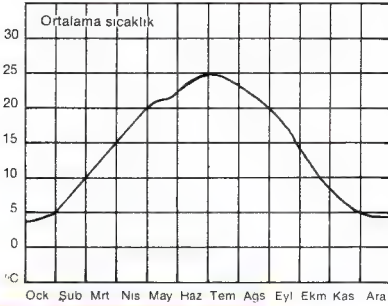


Çubuklu Şemalar ve Grafikler:
Çubuklu şema, bir dizi kolondan ya da çubuktan oluşur. Her kolunun uzunluğu temsil ettiği şeyin tutarına göre değişir. Örneğin bir mağaza, yıl boyunca satılan manto ve pardösülerin sayısını ay ay belirtmek istiyor. Satış figürleri aşağıda gösterilmiştir.

Aynı bölgede bulunan meteoroloji istasyonu da, her ayın ilk günündeki ortalama sıcaklıkları not ediyor. Bu figür de aşağıdaki grafikte gösteriliyor. Ortalama dereceler belirtildikten sonra,

araları düzgün bir çizgiyle birleştirilebilir. Bu, bizim yılın herhangi bir zamanındaki ortalama sıcaklığı rahatlıkla okuyabilmemizi sağlar. Örneğin ekim ayı ortasındaki ortalama sıcaklık 12° dir.

Grafikle çubuklu şemayı yan yana inceleyecek olursak, manto ve pardösülerin en çok soğuk aylarda, en az da sıcak aylarda satıldığını kolaylıkla gösterebiliriz. Sıcaklık ve manto-pardösü satışları arasındaki ilgiyi de, bu şekilde hemen koyabiliriz.

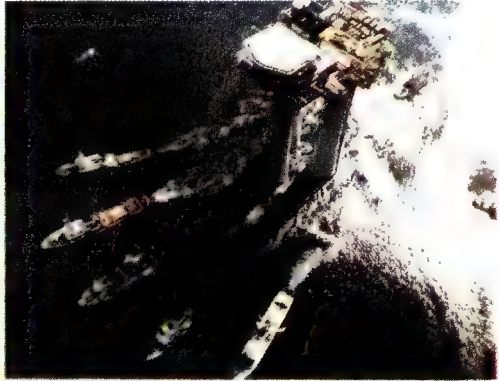


KAÇ ARABA?

Pazartesi	570
Salı	520
Çarşamba	530
Perşembe	510
Cuma	620
Cumartesi	320
Pazar	15

Yedi ayrı motor, tankeri değişik yönlere doğru çekiyor. Her motorun çekim kuvveti, değişik vektörlerle gösterilebilir. Sonra

bu vektör şeması, toplam kuvveti ve kuvvetin yönünü bulmak için kullanılabilir. (Altta)



Benzer ve Farklı

GENELDE, maddeleri ortak benzerlikler taşıyan gruplara ayırmak çok yararlı olur. Örneğin, bir spor merkezindeki çalışmaları planlarken, insanları uğraşmak istedikleri sporlara göre ayırabiliriz.

Bu gruplara, matematikte "küme" adı verilir. Her kümeye ayrı bir isim konabilir:

{futbol oynamak isteyenler} = X

{tenis oynamak isteyenler} = Y

{basketbol oynamak isteyenler} = Z

{yüzme isteyenler} = T

Bütün bu insanların başka bir ortak yönü de var: Hepsi spor merkezine geliyor. Yani, bir başka küme daha var: spor merkezine giden herkes = E

Bu kümeye **evrensel küme** denir. Çünkü problemimizdeki herkesi içine alıyor.

Düşünebileceğimiz her şey bir küme içinde yer alabilir. Eşyalar, insanlar, hatta düşünceler bile kümelere ayrılabilir. Kümeler ve evrensel küme, her probleme uygun bir şekilde seçilebilir. Bir müdür, okuldaki çocukları şu şekilde kümelere ayırabilir:

{bütün kızlar}

{bütün erkekler}

{birinci sınıftakiler}

{1.20 metreden kısa olanlar}

Bu müdüre göre, okuldaki bütün çocuklar evrensel kümeyi oluştururlar.

Bir kütüphanecinin evrensel küme, kütüphanesindeki kitapların tümü olabilir. Bu kitaplar roman, araştırma, çeviri, çocuk kitapları vb. kümelere ayrılabilir.

Havaalanında çekilen bu fotoğraf, 25 insanı gösteriyor.

Her insanın özellikleri ayrı; bazıları erkek, bazıları kadın, bazılarının bavulu var, bazılarının yok, vs. Daha küçük resimlerde bu insanlar değişik kümelere ayrılmış. (Sağda)



Resimdeki insanların tümünün oluşturduğu evrensel küme, daha küçük kümelere ayrılabilir.

A = {balkondaki insanlar}

B = {aşağıdaki insanlar}

C = {oturanlar}

Oturan insanların tümü (C kümesi) aşağı katta, yani hepsi aynı zamanda B kümesinin de içinde. C kümesi, B kümesinin alt kümesi oluyor. Çünkü bütün elemanları, aynı zamanda B kümesini de oluşturuyor. Bu şu şekilde yazılabilir: C ⊆ B

(Üstte)

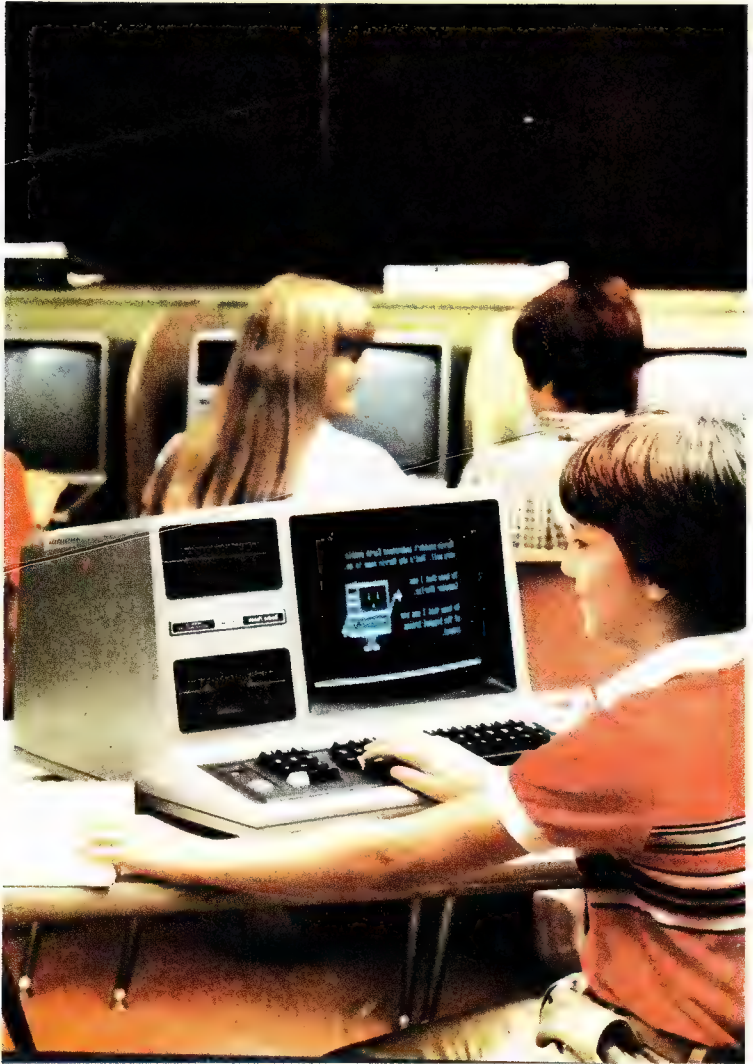




D kümesi, merdivenin yanındaki insanları belirtiyor. Ana resimden görebileceğiniz gibi D kümesinde yedi kişi var, ikisi üst katta, beşi alt katta. Bunlar da kendi aralarında E kümesini oluşturuyorlar. E, hem B'nin hem de D'nin alt kümesi. Yani B ve D'nin kesişimi: Bu da, şu şekilde yazılıyor: $E = D \cap B$. (Solda)

Merdivenin yanındaki insanlarla balkondakılar, F kümesini oluşturuyorlar. A ve D, F'nin alt kümeleri, yani F kümesi A ve D'nin "bileşimi". Bunu şöyle de gösterebiliriz: $F = A \cup D$ (Sağda)





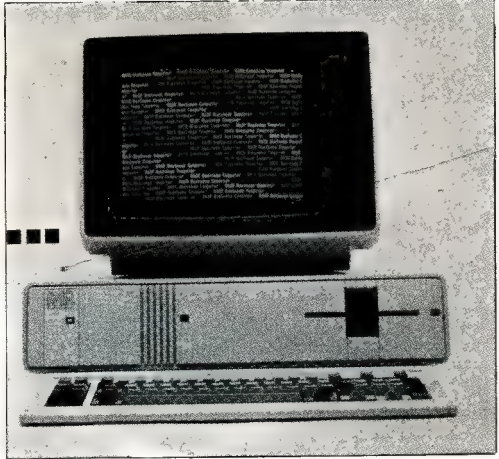
Akış Diyagramı

BİLGİSAYARDA her şeyin adım adım olduğunu daha önce belirtmiştik. Bu adımları akış diyagramında gösterebileceğimizi de açıklamıştık. Akış diyagramı, bir fonksiyonu uygulamak için gereken işlemlerin şematik bir biçimde gösterilmesidir. Bilgisayarın, işlemleri ne şekilde, hangi sırayla yapacağını belirtir. Bu diyagram ne kadar detaya inerse, o kadar açık yazılmış demektir. Yani biz bir işlemde her türlü olasılığı düşünüp, ona göre bir diyagram yaparsak, işlemi bilgisayara o derece açıklamış oluruz. Bilgisayar hiçbir zaman kendi kendine düşünemediği için, önceden açıklanmamış bir durumla karşılaşınca işlemi sürdüremez.

Problemın Kavranması: Akış diyagramı, bize bilgisayarın bir problemi çözerken yürütmesi gereken mantığı gösterir. Yani aslında herhangi bir problem için program yazmadan önce yapılması gereken bir şeydir. Özellikle çok karmaşık ve uzun programları yazarken, akış diyagramları bize çok yardımcı olurlar. Bu tip bir program yazmak isteyen bir kişi, eğer zamanı tasarruf etmek amacıyla akış diyagramını çizmeden işe girişirse, sonradan yapabileceği bir yanlış bulup, işin içinden çıkabilmek için çok daha fazla zaman kaybeder.

Bu nedenle en basit programları yazarken bile, önce akış diyagramını çizmeli, sonra programımızı ona göre yazmalıyız. Bu şekilde, programı yazmadan önce, kafamızda her şey açık olur. Her türlü olasılık hesaplanmış, ona göre ayrı yollar sapmamıştır. Bize yalnızca diyagramdaki eylemleri kullanacağımız dilin komutlarına göre, çevirerek yazmak kalır.

Yabancı dil programı eklenen bilgisayarlar, eğitimin de hizmetindedir. (Karşı sayfada)



Bir mikro bilgisayar. (Üstte)

Bunun için, önce problemi kavramamız gerekir. Yani bu problem bizden ne istiyor, sonucu bulmak için ne gibi işlemler yapmamız gerekir, vs. gibi soruların yanıtlarını bulmamız gerekir. Ancak bu şekilde o problem hakkında bir mantık yürütebiliriz.

Kendimize göre bir mantık koyduktan sonra, akış diyagramını çizebiliriz. Akış diyagramını çizerken, dikkat etmemiz gereken bazı noktaları yineleyelim.

Veriler: Verilerin çok iyi bir şekilde saptanması gerekir. Ne istenenden fazla, ne de az olmalıdır. Fazla olmasının, sonuç çıkmasına bir etkisi olmaz. Ama, bilgisayarlarla yapılan çalışmalarda işlemi en kısa yoldan yapmak, sonuca en kısa yoldan varmak önemlidir. Bir bilgisayarın ana işlemcisinin kapasitesi ne kadar fazla olursa olsun, bir anda çok fazla sayıda işlem yapılabilmesi açısından her

bir programın olabilecek en az yeri kaplaması gereklidir. Bunun için gereksiz işlemlerden kaçınmamız gerekir. Çünkü bir programın az yer kaplaması da önemlidir. Akış diyagramları da programların ön aşaması olduğu için, bu kısaltma işine oradan başlamamız gerekir.

Eksik olması ise, çok daha sakıncalıdır. Bir bilgisayar, eksik verilerle hiçbir zaman bir sonuç çıkaramaz. Onun için problemi kavramak, gereken verileri kesin olarak saptamak zorunludur. Veriler saptandıktan sonra, asıl fonksiyonun yazılmasına geçilir.

Fonksiyon: Fonksiyona geçtiğimizde, daha da dikkatli olmamız gerekir. Çünkü fonksiyon yazılımlarında çok daha fazla yanlış yapılır. En ufak bir belirsiz nokta bırakılmamalıdır. Her zaman belirttiğimiz gibi bilgisayar kendi kendine düşünemez. O, ancak kendisine verilen komutlara göre hareket edebilir. Eğer belirsiz bir nokta kalırsa, ne yapacağını bilemez; bu nedenle de işlemi durdurur.

Ayrıca değerlendirme nokta

Bilgisayar donanımlı bir film yıkama makinesi. (Sağda)

ları, değişik yönler ve tutar iyi saptanmalıdır. Yanlış hazırlanmak bir program, bilgisayarın sonsuza kadar aynı işlemi yinelenmesine neden olabilir. Bunu engellemek için akış diyagramları tekrar tekrar kontrol edilmeli, üzerinde, üzerinde uygulama yapılmalıdır. Zaten bir diyagramın zayıf noktalarını yakalamanın en iyi yolu işlemi üzerinde uygulamak, birkaç örnek çözmektir. Bu şekilde daha öncesinden gözden kaçan belirsiz noktalar, eksiklikler ve yanlış komutlar yakalanabilir. Böylece bilgisayarın işlemi durdurması, ya da sonsuza kadar yinelenmesi önlenmiş olur. Bu ön aşamada kontrol etmek için ayıracığımız birkaç dakika sonradan bizi saatlerce düşündürüş için yanlış tarafını aramaktan kurtarabilir. Bunun için, özellikle ilk çalışmalarda zamandan tasarruf etmeye çalışmamalı, programı yazıp bilgisayara vermeden önce tekrar tekrar kontrol etmeliyiz.

Sonucun alınması: Fonksiyon kısmı bittikten sonra akış diyagramının son bölümüne geçilir. Bu bölümde sonuç çıkarılır ve herhangi bir-terminalden alınması sağlanır. Çıkışta, yalnızca sonucu almak yeterli olmayabilir. Örneğin bir işlemi değişik koşullarda, değişik verileri uygulamak istemiysek, o verileri de almamız gerekir. Bu şekilde sonuçlar karışmamış olur; ve yine programda herhangi bir yanlış varsa daha kolay bir şekilde açığa çıkar. Çünkü bazen bir programda yanlışlar olsa bile, sonuç çıkabilir. Yalnız bu sonuçlar mantık dışı olabilir. Ama eğer verilerle sonuçlar birlikte çıkarsa, kısa bir kontrole yanlış bir yer olup olmadığı anlaşılabilir. Bu yanlışlıklar üzerinde çok duruyoruz. Çünkü genelde hiçbir program ilk yazıldığında doğru sonuç vermez. Özellikle uzun, karışık programlarda, mutlaka gözden kaçan birtakım yerler





vardır. Bir programcı, sürekli bu tip yanlışları kontrol etmek, bulmak durumundadır.

Sonuçla birlikte alınacak çıkışlar saptandıktan sonra çıkış diyagramı bitirilir. Son kontroller de yapıldıktan sonra, sıra program yazmaya gelir. Biz şimdilik program yazma kısmına geçmeyip, birkaç örnek çözelim. Önce çıkış diyagramında kullanılan değişik şekillerdeki kutucukların ne anlama geldiğini açıklayalım. Bu kutucuklar, başka başka şekillerde de kullanılabilirler. Bizim kullanacaklarımız, en yaygın olanlardır. Bunların bir kısmı daha önceki diyagramında kullandığımız şekillerin aynısıdır. Ancak birkaç şekil, daha değişiktir.

Başlangıç: Her program, başlangıç komutuyla açılır. Akış diyagramında da bu komut özel olarak gösterilir. Bu komut, bir elips içine alınarak yazılır:

BAŞLANGIÇ

Girişler: Başlangıç komutundan sonra verilerin girişi yapılır. Girişler, bir paralel kenar içine yazılarak gösterilir:

A'NIN
GİRİŞİ

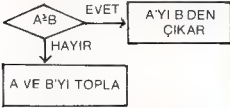
Fonksiyonlar: Girişler tamamlandıktan sonra, fonksiyonların yazılmasına geçilir. Her bir fonksiyon, dikdörtgen içine alınarak gösterilir:

A VE B'Yİ TOPLA

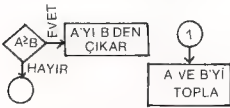
Günümüz mikro bilgisayarların çoğu, çok renkli çizimler de yapabilmektedir. (Yanda)



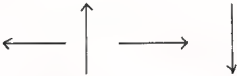
Değerlendirme: Kullanılan fonksiyonlarda, değişik durumlar için değişik yollar çıkabilir. Örneğin bir problemde, kullanılacak olan sayı önceden saptanmış bir değerden küçükse başka bir işlem, yapılabilir. Bu koşullarda değerlendirme fonksiyonu kullanılır. Bu fonksiyon bir eşkenar dörtgen içinde gösterilir:



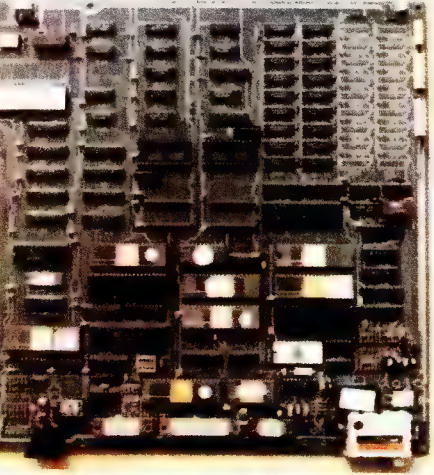
Bağlama: Bazı uzun ve karmaşık programlarda, akış yönü çok karışık olabilir. Bunun için bağlayıcılar kullanılır. Bu bağlayıcıların yardımıyla akış diyagramı üzerindeki fonksiyon ve değerlendirmeler daha açık ve net bir biçimde görülebilir. Bağlayıcılar, küçük bir dairenin içindeki numaralarla gösterilir. Bu numara herhangi bir fonksiyonun arkasında olabilir. Yani herhangi bir fonksiyonun gideceği yeri gösterebilir. Sonra aynı numara gidilecek yere de konur. Bunu bir örnek ile gösterecek olursak:



Akış yönü: Akış diyagramında akış yönü oklarla belirtilir. Oklar, akış yönüne göre bir sonraki kutucuğa doğru çizilir.



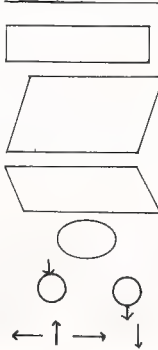
Çıkış: Çıkışlar, sonuçta terminalden istenecek olan şeyler belirlenir. Çıkışın gösteriliş şekli, girişin gösteriliş şekliyle aynıdır:



Bir mikro bilgisayarın ana ünitesi. (Üstte)



Bitiş: Her programın bir başlangıcı olduğu gibi, bir bitiş, yani SEMBOL



sonu da olmalıdır. Program sonunun gösterilmesiyle de, program başlangıcının gösterilmesi aynıdır:

Bakın bu anlatıklarımızı tek bir cetvelde gösterecek olursak:



FONKSİYON	ANLAMI
Programdaki fonksiyonu gerçekleştiren bir komut dizisi.	
GİRİŞ/ÇIKIŞ	
Giriş/çıkış elemanlarının, herhangi bir fonksiyonu (veri girişi yapmak, sonuçları yazıya geçirmek, vs.)	
DEĞERLENDİRME	
Değişen koşullara göre ayrılan akış yönlerinin gösterilmesi.	
TERMINAL	
Bir programın başlangıcı, bitiş	
BAĞLAYICI	
Akış diyagramının bir yerinden başka bir yerine giriş ve çıkışlar.	
AKIŞ YÖNÜ	
İşlemin gittiği yön	

Programlamaya Giriş

PROGRAM, disk, disket gibi belleklerde saklanan bir komut dizisidir. Programı, bir yemek tarifine ya da bir yapı işi kurallarına benzetebiliriz. Yumurtta pişirmeyi ele alalım ve bu tarifi adım adım yazalım.

1. Yumurtaları sahana koy
2. Üzerini kaplayacak kadar su ekle
3. Suyu kaynatmaya başla
4. Kaynayınca altını kıs
5. Beş dakika bekle
6. Yumurtaları soğuk suya geçir

Bir de duvar örme şekillerinden birine bakalım:

1. Temel kaz
2. İçine duvarı dik tutacak sağlam direkler çak
3. Altına beton dök
4. Bir sıra tuğla yerleştir
5. Tuğlaların arasında ve üzerine tuğla koy
6. Tuğla yerleştirip aynı işlemleri yinele
7. Duvar bitince üzerine sıva çek.

Bu iki örneğe birlikte göz atalım. İkisinin de birçok ortak yönü var. Birincisi, her madde olabildiğince kısa yazılmış ve her madde ayrı ayrı numaralanmış. İşin doğru bir şekilde yapılabilmesi için bu sıranın izlenmesi gerekiyor. Ayrıca her iki örnek de birden fazla kullanılabilecek bir şekilde yazılmış. Bütün bu benzerlikler, bir bilgisayar programı için de söz konusu. İşte bu yüzden programlar bellek ünitesinde saklanır ve gerektiğinde yeniden kullanıma hazırdır. Bildiğiniz

Mikro bilgisayarların içinde bulunan devrelerin maketleri (Sağda üstte).



gibi bu programları, programcı dediğimiz kişiler hazırlar.

Programcı, bilgisayarın ne tür bir iş için kullanılacağına karar verir ve bu işin nasıl yapılacağını bilgisayara adım adım anlatan program dediğimiz komutları yazar. Bu işi, tıpkı yumurtanın pişirilmesi tarifinde olduğu gibi yapar.

Ancak bu komutlar, öyle herhangi bir şekilde yazılamaz. Bilgisayarlar çalışırken, ne istediğini kesin olarak belirtmek zorundasınız. Belki şimdi şu anda, ben isteklerimi her zaman kesin olarak belirtirim” diyenleriniz çıkabilir. Size şunu söyleyelim, bizim bir bilgisayar olduğumuzu varsayın ve bizden iki sayının toplamını isteyin.

Buna şöyle bir karşılık gelebilir: “Evet, 8 ve 3 rakamlarını toplayın.”

Peki ama bunları niye toplayacağız?

“Doğaldır ki, birbirine!”

Ancak bunu bize başta belirtmediniz, değil mi? Bu rakamları başka bir yere eklememiz koşulu da olabilirdi. Örneğin 5 rakamını. Bunu bizim düşünebileceğimizi söyleyebilirsiniz. Ama biz de size, defalarca söylediğimiz bir şeyi yineleyeceğiz. Bilgisayar, hiçbir zaman kendi kendine düşünmez!

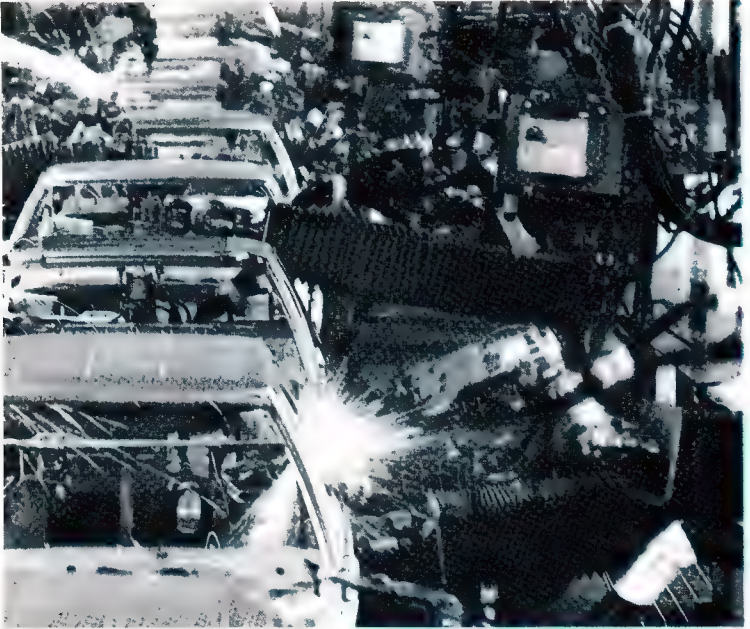
Peki, sonra ne yaparsınız? Eğer yanıtın gelmesini bekeleyecek olursanız, epey bir süre bekleyeceksiniz demektir. Çünkü bize, sonucu söylememiz gerektiğini de belirtmediniz. Bir bilgisayar, sadece kendisine söyleneni yapar. Ne fazlasını, ne de eksikliğini.

Bir programcının bu tip yanlışlar yapmaması için, sık sık program çalışması yapması gerekir. Yani deneyim kazanmak çok önemlidir. Siz de başlarda bu tip yanlışlar yapabilirsiniz sakın üzülmeyin. Bir süre sonra siz de alışırsınız.

Bir başka önemli nokta da, **komutların sırasıdır**. Bu sıranın kesinlikle doğru olması gerekmektedir. Örneğin, yumurtta pişirme tarifinde, ikinci maddeyle üçüncü maddenin yerini değiştirsek ne olur? Herhalde kaynamış yumurtalar yerine, yanmış yumurtalarımız olur.

Ayrıca, verilen bütün bilgilerin kesinkes doğru olması gerekir. Ufak gibi görünen şeylerin bile, örneğin sinemanın yerini öğrenmek isteyen birine; caddeye çıkınca sola dönmesi yerine sağa dönmelerini söylerseniz ne olur? Herhalde zavallı adam, sinemaya birkaç yüz metre uzaklıkta umutsuzca dolaşıp durur.

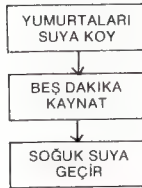
Yani, bir programın her



Bilgisayar bir makine değil... Ya da, bilgisayarın makine kısmı kendi başına çamaşır makinesi ya da ütuden farksızdır... Çünkü bilgisayarla çalışmada anahtar, bu makinelerin "hayat bulmasını" sağlayan temel faktör, programcılık ve destek hizmetleridir. (Üstte)

adımının doğru olması gerekir. Küçük bir yanlışlık, polisin masum bir adamı tutuklamasına, kontrol sırasında bir roketin havaya fırlamasına ya da bir fabrikadaki tüm kimyasal maddelerin birbirine karışmasına yol açabilir. Bu gibi örnekleri kolaylıkla çoğaltabiliriz. Bu yüzden programcı, büyük bir dikkatle adımlarını belirler ve bunların kesinlikle doğru sırada ve doğru biçimde olmasına çalışır. Akış programı bu sırada devreye girer.

Yine biz yumurta tarifine dönelim. Bunun akış diyagramını çizsek:



Dikkatinizi çektiyse, çok kısa adımları bir araya topladık. Böylece çok fazla kutucuğumuz olmadı. Bu çok kolay bir tarif olduğu için, programımızı kolaylıkla çizebildik. Şimdi bunu biraz değiştirelim. Sahanımızın çok küçük olduğunu varsayalım.

Öyle ki, içine sadece tek yumurta alabiliyor. Bu kez, tarifi birkaç kez yinelememiz gerekiyor. Bunun için de, tarifimize yeni bir madde ekleyeceğiz.

7. Birden fazla yumurta pişirmek için, yukarıdaki maddeleri yinele.

Akış diyagramında, buna bir ok daha ekleyerek yapabiliriz.





Uzay merkezlerinde kullanılan bilgisayarlar da özel nitelikler araniyor. (Ustte)

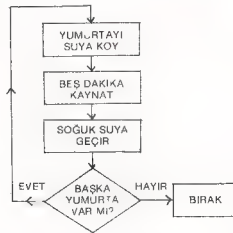
kacağız. Burada bize gerekli olan, deęerlendirme komutudur.

Yemeęimizde bozuk yumurta olup olmadıęından emin olmamız gerekir. Eęer varsa, yerine bařka bir yumurta koymalıyız.

Bilgisayar dilinde, buna **tur** denir. Yumurtalarımız bitene kadar, adımları yineler durur. Yumurtanın bir ya da bin olması önemli deęildir. Program bir kez belleęe alındı mı, aynı komutlar tekrar kullanılabilir.

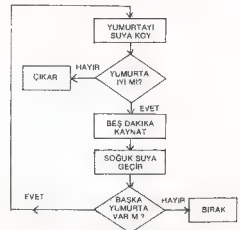
Ama bu programda bir eksiklik var. Bize yalnızca adımları yinleyeceęimizi belirtiyor. Yumurtalar bitince ne yapacaęımızı göstermiyor. Eęer **diya**gramımızın kusursuz olmasını istiyorsak, buna bir çözüm bulmak zorundayız.

Tarifin sonuna her gelişimizde bir soru soracaęız; bařka yumurta var mı? Eęer yanıt evet olursa, bir ok bizi bařa götürüyor. Eęer yanıt hayır ise, bira-



Deęerlendirme, bilgisayar çalışmalarında önemli bir yer tutar. Ve genelde **bir bilgisayar programında da en azından birkaç tane deęerlendirme komutu vardır.**

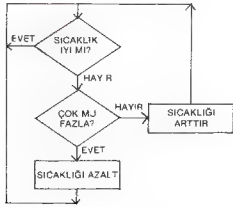
řimdi tarifimizde bir deęerlendirme daha yapalım. Bozuk yumurtaları nasıl anlarız? Suyun üzerinde yüzerler, deęil mi?



Bu kez deęerlendirme sembolü, bize programı bitirmeden geri dönme olanaęı veriyor. **HAYIR** oku, bizi taze bir yumurtayla işe yeniden başlamak için bařa götürüyor. Bu arada da **EVET** oku da programı eskiden olduęu şekilde sürdürebilmemizi saęlıyor.



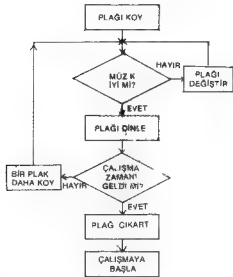
ARTIK akış diyagramlarında değerlendirme sembollerinin, adımların ne zaman yinelenmesini, ne zaman atlanacağını ve önümüzdeki iki yol çıktığında hangisini seçeceğimizi gösterdiğini biliyoruz. Örneğin bir bilgisayarın bir tabir-kada kullanılan kimyasal bir alaşımları kontrol ettiğini düşünelim. Sürekli sıcaklığı ölçüp, istenmeyen sıcaklıkları değiştirme işini yapıyor. Sıcaklığı azalttıktan ya da arttırdıktan sonra bile, kontrol etmeyi sürdürmek zorunda:



Bu tip bir tura 'sonsuz tur' adı verilir. Çünkü sistem kapatılmadığı sürece hiçbir zaman yineme işlemini yarıda bırakmaz.

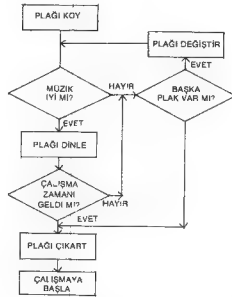
Evet, artık düşünebildiğiniz her olayın akış diyagramını çizebilirsiniz. Neden denemeyesiniz?

Plak dinlemeyi ele alalım:



Az ötede arı gibi çalışan robotlara komuta eden beyin bu işte. Hem enerji hem de komut veriyor. (Solda)

Bu oldukça iyi oldu, değil mi? Ama yine de bir şeyi unuttuk. Eğer plakları beğenmezsek sürekli değiştirecek miyiz? İstemediğimiz halde, bir sonsuz tur yapmış olduk. Bu yanlış düzeltmek için bir tane daha değerlendirme sembolü koyacağız:



Bu konuda artık deneyim kazandık sayılır. Şimdi de diğer sembolleri kullanarak iki sayıyı toplayalım:



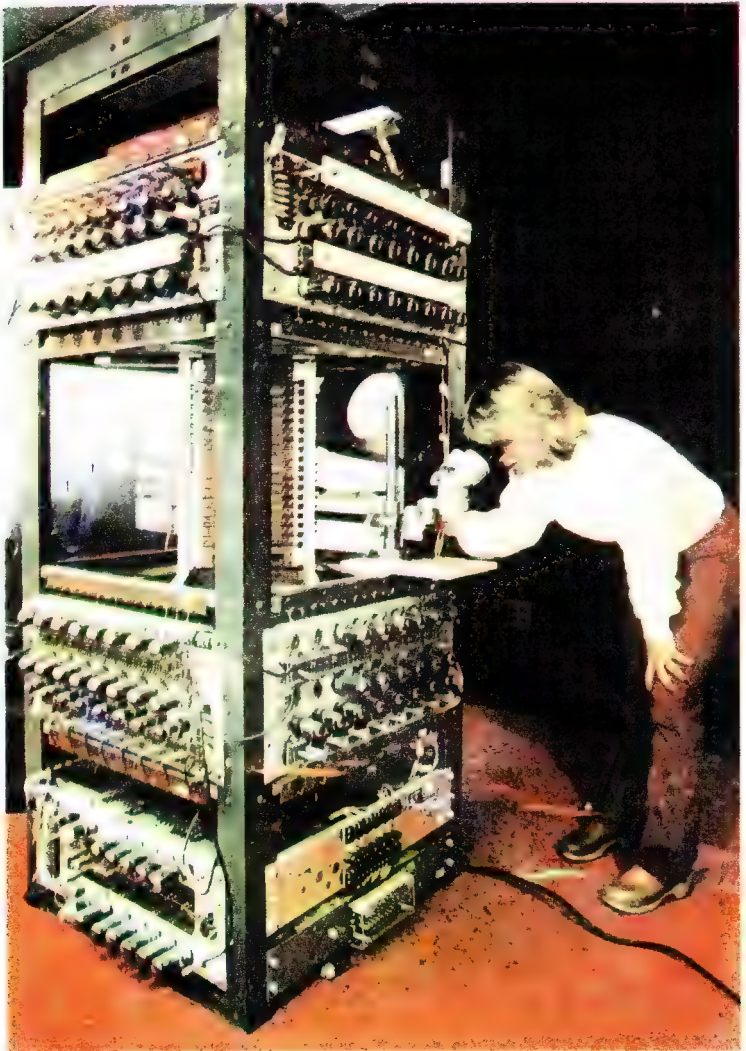
Gayet kolay, değil mi? Eğer üçüncü bir sayıyı daha olsaydı,

ne yapardık? Onu da toplama eklerdik. Dördüncüsü, beşincisi olsa, yine aynı şeyi yapardık. Elimizde bir sürü sayı olsa, yine aynı şeyi yapardık. Doğaldır ki, sayılar bitene kadar. Bunun akış diyagramı da şöyle olurdu:



Şimdi de bir sınıf düşünelim. Bu sınıftaki çocukların notlarının ortalamasını alalım. Bunu yapmak için de şöyle bir diyagram hazırlarız:

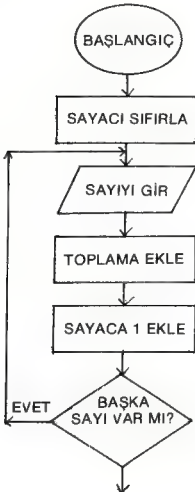




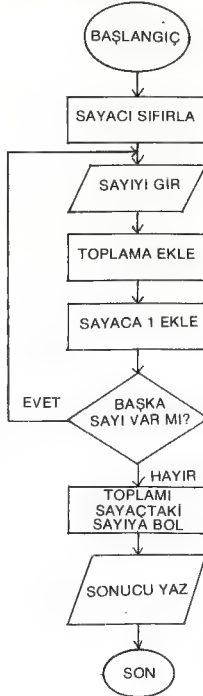
Ama biz bu diyagramda da bir yanlış yaptık. Biz bu verilerle ortalamayı bulamayız. Çünkü kaç notun ortalamasını alacağımızı belirtmedik. Yani toplamı kaç bölceğimizi bilmiyoruz. Ayrıca biz bu akış diyagramını öyle yapmalıyız ki, tekrar tekrar kullanılabilirsin. Başka tip notlar, başka sınıflar kullandığımızda da işimize yararsın. Bunun için toplamı kaç bölceğimizi ayrı bir kutuda belirtmemiz gerekiyor.

Elimizde pek çok sayı olduğunda ve biz bunların kaç tane olduğunu bilmek istediğimizde ne yaparız? Sayarız, değil mi?

Akış diyagramında da buna benzer bir şey yapma olanağımız var. Bu işi, 'sayacı' adı verilen bir tür bellek yardımıyla yapabiliyoruz. Sayacı akış diyagramının içine yerleştirdiğimizde, o bizim yerimize notları o sayıyor. Biz, sayaca '0' sayısını koyuyoruz. O, her toplama işlemi yapıldıktan sonra, kendisine '1' ekliyor. Bu şekilde her toplama sonucu, sayacın üzerinde kaç not toplandıysa o sayı bulunuyor.



Ve son notu da toplama eklediğimizde, sayacın elimizdeki sayıların tutarını bize söyleyebilmesi gerekiyor. Bize de toplamı sayıdaki sayıya bölme işlemini yaptırmak kalıyor.



Bu akış diyagramı, programcılar çizdiği diyagramlara oldukça benzedi. Programcıların programlarını yazmadan önce neden akış diyagramı çizdiklerini anladınız mı?

Mimar, evin planını çizmeden önce bir taslak çıkarır. Yazar, romanını yazmadan önce bir plan belirler. Ressam da bir şeyin resmini yapmadan önce onun üzerinde çalışmalar yapar. Peki, bir

programcı programını nasıl yazar?

Önce, çizdiği akış diyagramının mükemmel olduğuna karar vermesi gerekir. Sonra da bu program için kullanabileceği en uygun program dilini seçer.

Size, programlama dillerinden daha önce söz etmiştik. Şimdi daha detaylı inceleyelim. Programlama dilleri çok çeşitlidir. Bunları İngilizceye ya da Fransızca'ya benzetebiliriz. Biz bu yabancı dilleri hem konuşup, hem de yazma olanağında edebiliriz. O dilden konuşan insanları anlayabilmek, onlara karşılık verebilmek ve o dilden yazılmış kitapları okuyup anlayabilmek, gerektiğinde yazabilmek için öğreniriz.

Ancak, programlama dilleri bilgisayar dilleridir. Onlarla anlaşmak için kullanılırlar. Yalnızca yazılı olarak işlevlerini sürdürebilirler. Şimdilik yalnızca yazılı girişten değil de, konuşmadan da anlayan bilgisayar türleri yeterince geliştirilmemiştir. Ama bu dillerin işlevleri, diğer yabancı dillerle aynıdır. Bilgisayar, bizim dilimizden anlamaz. O yüzden bizim onun anlayacağı bir dili kullanmamız gerekir.

Programlama Dilleri

ÇEŞİTLİ programlama dilleri vardır demiştik. Bunlara örnek olarak FORTRAN, COBOL (kobol okunur) - BASIC (beyzik okunur) - ASSEMBLER (assembler okunur) dillerini gösterebiliriz. Bunun nedeni, bilgisayarların çok çeşitli işlerde kullanılmalarıdır. Her dil, başka türden işler için kullanılır. Bu dil de, o işi en iyi açıklayan dil olduğu için seçilir.

Nasıl ki, bir marangoz, işinde kendisine en çok yarayacak aletleri kullanırsa; örneğin çivi çakmak için çekiçi, çıkartmak için de kerpeteni kullanırsa, bilgisayar dilleri de en çok yararlı olabilecek işlerde kullanılır.

Bilgisayarlar, gelişmiş ülkelerde evlerde de kullanılıyor. (Sağda)

Bu işleri şöyle örnekleyebiliriz. FRTRAN, çoğunlukla mühendislik işlerinde kullanılır. Çünkü bu dil, bilimsel konular için hazırlanmıştır.

Bankalarda ise, en çok COBOL kullanılır. COBOL, muhasebe işlerine uygun bir dildir.

BASIC, genelde küçük sistemlerde kullanılan bir dildir. Çünkü öğrenimi ve kullanılması çok basittir ve bu yüzden çok basit programlar rahatlıkla yazılabilir.

ASSEMBLER, pek kullanışlı bir dil değildir. Hem çok uzundur, hem de çok karmaşıktır. Yalnız, bilgisayarın tam olarak çalışmasının kavranması açısından öğrenilmesi gereken bir dildir.

Anında Çeviri

HER bilgisayarın bütün dilleri bilmesi gerekmez. **Simultane** çevirmenin ne iş yaptığını biliyorsunuz, değil mi? Herhangi bir dilde konuşan birinin söylediklerini dinleyip anında başka bir dile çevirir. Bu şekilde, o insanın konuştuklarını herkesin anlamasını sağlar.

Sözgelimi, Birleşmiş Milletlerde, herhangi bir ülkenin temsilcisi konuşmaya başlayınca, bu çevirmenler konuşmayı diğer dillere çevirir. Anak bu şekilde her ulusun söyledikleri ötekiler tarafından anlaşılabilir.

Bilgisayarın içinde bu tip bir çevirmen vardır. 'Çevirici' adı verilen bu program, CPU'nun belleğinde sürekli olarak saklanır. Programın işlevi, kendisine gelen çeşitli dillerdeki programları bilgisayarın mekanizmasına uygun bir dile çevirmektedir.



Bilgisayarın belleklerinde bilgilerin nasıl saklandığını anımsıyorsunuz, değil mi? Hepsi ikilik düzende saklanır. Programcı, programını ikilik düzende yazmaz. Onu için, daha önce sözünü ettiğimiz dillerden birini kullanır.

Çeviricinin işi, burada başlar. Programdaki her komutu inceler, ikilik düzene çevirir ve bilgisayarın belleğine aktarır. Burada program, CPU'nun kullanımına hazır bir şekilde durur.

Bir program ikilik düzene çevrildikten ve belleğe aktarıldıktan sonra, şu şekildedir:

100010	10111000
100011	10001101
100100	00111011
100101	01110000
..	.
..	.

Sağ taraftaki sayılar, komutların ikilik düzene yazılışlarıdır. Sol taraftakiler ise, bu komutların adresleridir. Adresler konuşuna da önce bir parça değinmiştik. Bu adresleri, o komutların evleri olarak açıklayabiliriz. Bellekte bir sayı ya da karakter saklayan her birimin adresi vardır.

Aylarca çalışarak elde edilebilecek bir bilgi, bilgisayarda birkaç saniyelik bir zamanda karşınıza çıkar. Bunu ekranda görebileceğimiz gibi, işitsel olarak da dinlememiz olasıdır. (Sağda)

CPU, onları ancak bu adreslerinden yararlanarak bulabilir.

Program komutları da bellette saklandığına göre, onların da adresleri olması gerekir:

ADRES	KOMUT
34	SAYACI SIFIRLA
35	SAYIYI OKU
36	TOPLAMA EKLE
37	SAYACA 1 EKLE

Adresteki sayıların sıralı olmasına dikkat ettiniz, değil mi? Sokaklardaki ev numaraları gibi, onların da numaraları belli bir düzen içinde. Arka arkaya bulunan komutların adres numaraları arka arkaya. Bu şekilde çevirici CPU'ya ilk komutun adresini verdiğinde, CPU'nun ikinci komutu bulması için verilen adres numarasına 1 eklemesi yeterlidir. Böyle birer birer ekleyerek üçüncüyü, dördüncüyü, beşinciye ve diğerlerini de kolaylıkla bulabilir. Tıpkı akış diyagramında yer alan sayacın yaptığı gibi.

Şimdi öğrendiklerimiz bir kez gözden geçirelim. Siz, bir programcısınız ve bir program yazmak istiyorsunuz.

Önce akış diyagramını çizersiniz. Sonra, bu işi en iyi açıklayabilecek programlama dilini seçersiniz. Programı yazıp, belleklerden birine yerleştirirsiniz. Sonra çevirici sizin programınızı alır, ikilik düzene göre yeniden yazar ve ana belleğe aktarır. CPU kullanana dek, program orada kalır.

CPU, bellekteki programı çeviricinin yardımıyla bulur. Çevirici, ona programın adresini söyler. CPU, programın başlangıç adresini öğrendikten sonra, diğer komutları da bu adreslere birer birer ekleyerek bulabilir.



Çok güzel, şimdi bunun akış diyagramını da çizebiliriz.



CPU ve ana belleğin içine girmişken, akış diyagramları ve programlardan bir süre için ayrılmış, başka şeylere bakalım.

CPU'nun, bir bilgisayar sistemindeki en önemli ünite olduğunu belirtmiştik. Bizim vücudumuzu bir bilgisayar sistemi olarak düşünersek CPU, beynin işlevini yüklenir. CPU ve ana bellek birlikte çalışırlar. Daha doğrusu CPU, işlevini sürdürebilmek için ana bellekteki bilgileri kullanır.

CPU'nun görevlerini, aritmetik fonksiyonlar, kontrol ve değerlendirme fonksiyonu olarak üç ana başlığa inceleyebiliriz. CPU, genelde iki bölüme ayrılmıştır. Birinci bölüm, aritmetik ve değerlendirme fonksiyonlarını üstlenir. İkinci bölüm de kontrol işini yapar.

Önce, aritmetiği ele alalım. Bu, bizim en başta sözünü ettiği-



Bilgisayarın levhalarından biri üretilirken görülüyor. Bilgi işlem iletişim için, cam levha içinde bakırın havası alınarak yerleştirilmesi sonucu, bilgisayarın ana maddesi olan iletken levhalar elde ediliyor. (Üstte)

miz Boolean Aritmetiğinden farklı bir şey değildir. CPU, kendisine verilen bütün problemleri, yalnızca bu aritmetiğin kurallarını kullanarak çözer. Çünkü içinde ikilik düzenden başka hiçbir şey yoktur aslında. VE, VEYA, değil ve diğer kapıları

anımsıyorsunuzuz, değil mi?

Şimdi, CPU'nun bu kapıları nasıl ayırt edebildiğini açıklayalım. CPU, kendi kendine kapıların hangisinin VE, hangisinin VEYA kapısı olduğunu anlayamaz. Artı olarak, hangi kapının ne iş gördüğünü de bilemez. Ona



bu konuda kontrol bölümü yardımcı olur.

Kontrol bölümü, beynimizde bize yürümeği gösteren bir bölüme benzer. Beynimiz bize önce bir ayağımızı, sonra da diğer ayağımızı atmamızı söyler. Ya da, dişimizi fırçalamamız için eli-

mizdeki fırçayı dişlerimizin üzerine tutup, elimizi ileri geri hareket ettirmemizi, konuşmak için dilimizle birlikte dudaklarımızı oynatmamız gerektiğini söyler. Kontrol bölümüne, hangi iş için neyi kullanacağını, ne şekilde, ne zaman ve nasıl kullanacağını belirtir.

Ana bellekle CPU, yan yana çalışırlar. Ama yine de ayrı bölümlerdedirler, yani aralarında doğrudan bir ilişki yoktur. Bu iki ünitenin arasındaki bağlantıyı, taşıtlar sağlar. Nasıl ki otobüsler insanları bir yerden bir yere götürüyorsa, taşıtlar da bilgiyi bir yerden bir yere götürürler. Taşıtların arasında bilgi taşıdığı yerlere istasyon adı verilir.

Bir bilgisayarda istasyon, bilgi giriş elemanları tarafından alınan bilgilerin taşıtlar tarafından diğer bölgelere götürülmek için durduğu yerdir. Ayrıca, taşıtlarla gönderilen sonuçların, çıkış terminallerine ulaşmadan önce uğradığı bir bölümdür. Bu giriş ve çıkış üniteleri, diskler, teypler ve yazıcılar ve VDU'lardır. VDU'ların, görsel çıkış üniteleri olduğunu bir kez daha anımsatalım. Yalnız, bu üniteler arasındaki iletişimi sağlayan taşıtlar, özel bir takım tellerden meydana gelir.

Taşıtlar, ana bellekle yan bellekler CPU ve çıkış terminalleri arasında sürekli bilgi taşır. Bu bakımdan ana belleği bir otobüs garajına benzetebiliriz. Bu garajda, taşıtlar sürekli olarak bilgi getirir, götürür, başka bölümlere aktarınlar.

Bu yüzden içi devamlı kalabalık ve karışıktır. Belleğin, kendisine verilen bilgileri nasıl sakladığını daha önce açıklamıştık. İsterseniz bir kez daha kısaca değinelim.

Belleklerde, çok sayıda küçük birimler vardır. Bu birimlerin her biri, bir tek sayı saklayabilir. Biz, bu birimlere 'bit' adını veriyoruz. Bu birimlerin birleştirilmesiyle oluşmuş her sayı ya da karakter, değişik şekillerde bir araya getirilmiş 0'lar ve 1'lerle gösterilir.

Bilgisayar, bu 0'ların ve 1'lerin neyi temsil ettiğini, verilen program yardımıyla öğrenir.

Program komutlarını kontrol bölümü okur ve gerekli işlemleri CPU'ya yaptırır.

Program belleğe aktarıldığı an, kontrol bölümü devreye girer. Komutları tek tek inceler. Her seferinde bir komutu alır, inceler, komutun ne yaptırmak istediğini öğrenir ve aritmetik-mantık bölümünden bu işi yapacak olan koşulları seçer. Onları yaptırır. Sonra yeniden belleğe geri döner ve programdaki bir sonraki komutu ele alır.

Eğer eldeki komut iki sayının toplanmasını istiyorsa, bunları aritmetik kısmına gönderir. İşlem orada yapılır. Eğer komut iki sayının karşılaştırılmasını istiyorsa, bunları bu kez de mantık kısmına gönderir. Değerlendirme işlemi de orada yapılır. Eğer komut daha önceden yapılmış bazı komutların incelenmesini istiyorsa, kontrol bölümü hemen başa dönüp bellekten istenilen komutları bulur; bunların yeniden yapılmasını sağlar.

Şimdi yeniden eskiye dönelim. Bitler, birleşip baytları oluşturuyorlardı. Baytlar, tamı tamına bir karakteri içine alabilecek kapasitede bellek birimleridir. Örneğin A, K, M, L, harflerinin her biri, tamı tamına bir baytı kapsayacak uzunluktadırlar. Ancak her sayı bir bayta sığmayabilir. Bir bayt onlar için çok kısa gelebilir. Böyle zamanlarda sayılar, yan yana birkaç baytın üzerine yazılır. Yan yana sıralanmış, belirli sayıda bayt gruplarına da, kelime adı verilir.

Biz, dilbilgisinde kelimeleri bir araya getirip cümleler kurarız. Bilgisayar da, kendi kelime'lerini birleştirip, 'kayıt' adı verilen bir bilgi grubu haline getirir.

Okulunuzda da bu tip kayıtlar bulunabilir. Öğrencilerin her birinin ayrı ayrı kaydı vardır. Bu kayıtlarda öğrencinin adı, soyadı, doğum tarihi veya yaşı, adresi ve varsa telefon numarası yazılır. Biz bu kayıtlara örnek olarak şiirin bir dizesini, ya da kitabın bir sayfasını da gösterebiliriz. Aynı şekilde, bellekte yer alan komutlar da birer kayıttır.



Okulunuzun öğrenci kayıtlarını bir arada tuttuğu gibi, dizelerin bir bütün olarak, ya da kitabın bütün sayfalarının birlikte saklandığı gibi, komutlar da bellekte bir arada saklanır. Bu kayıtlar grubunu da, bilgisayar dilinde 'dosya' adı verilir. Yani bellekteki tüm bilgiler, dosyalar halinde saklanır.

Bir dosya, çeşitli büyük - küçüklerde olabilir. Genelde, program ya da veri kayıtlarından oluşur. Yani bu bakımdan belleği, çeşitli türden dosyaları birlikte saklayan bir dolaba benzetebiliriz. Yalnız bu dolabın çok küçük bir dolap olduğunu ve içinde fazla miktarda dosyayı bir anda saklayamadığını hiçbir zaman unutmamalıyım. Çoğu dosyalar bu yüzden, yan bellek dediğimiz disk, disket, ya da teypelerde saklanmaktadır. Ancak bu şekilde belleğin içinde her zaman yer bulma olanağı var olabilir.

CPU, gereken dosyaları bu belleklerden ana belleğe alır ve orada gerektiği gibi kullanır. İş bittikten sonra da yerine koyması gerekmez. Çünkü, dosyayı ana

belleğe yalnızca kopya etmiştir. Bu, sizin kitabın bir bölümünü ders anlatırken tahtaya çekmeye benzer. Ve CPU, yeni bir dosya gerektiğinde, eski dosyayı bellekten siler; ve onun yerine yeni dosyayı geçirir. Bu da, sizin kitabın bir sonraki sayfasını tahtaya geçirmek için eskisini silmeye benzer.

Şimdi yine öğrendiklerimizin kısa bir tekrarını yapalım. Belleğin içinde bitler baytlar, baytlar da kelimeleri oluşturur. Kelimeler biraraya gelince kayıtlar, kayıtların birleşmesinden de dosyalar meydana gelir.

Şimdi biz yine bilgisayar programlamasına geçelim. Sanırsanız artık akış diyagramı hakkında aklınızda bir soru kalmamıştır. Akış diyagramlarının çizilmesi, programların yazılmasına göre oldukça kolaydır. Bazı programlar çok zordur ve bitirilmesi çok uzun zaman alabilir. Çok çeşitli diller bilen profesyonel programcılar, bu tip programların altından kalkabilecek biricik insanlardır.

Ama bazı program dillerinin

öğrenilmesi çok kolaydır ve bu dillerle çok basit programlar yazılabilir. Bu dillerden biri de BASIC'tir. BASIC, bugün dışarıda kullanılan ev için yapılmış bilgisayarlarda, okullarda, hatta birçok işyerinde kullanılan ve çok popüler olan bir dildir.

Yalnız, bu dillerin bizim için şöyle bir zorluğu vardır. Bunlardan hiçbirini henüz Türkçeye çevirmemiştir. Hepsinin komutları İngilizcedir. O yüzden biz de size bu komutları İngilizce vermek zorundayız. Yalnız anlamlarını ve ne işe yaradıklarını uzun uzun anlatacağız.

Basic Diline Giriş

DİĞER bütün programlama dillerinin olduğu gibi, BASIC dilinin de komutları İngilizcedir. Ancak komutlar, İngilizce anlamına çok benzer anlamlardadır. Bu yüzden, yalnızca anlamlarını vermek bile, bizi bu komutlar hakkında epeyce aydınlatacak.

Önce, en çok kullanılan komutlardan 'PRINT' komutunu inceleyelim. İngilizce 'print' kelimesi, 'basamak, basılı halde yazı ve resim çıkartmak' anlamına geliyor. Bu komutun ne için kullanıldığını da anlamışsınızdır her halde. 'PRINT' komutuyla bir sonucun elimize gelmesini, daha doğrusu sonucun bize söylenmesini sağlarız.

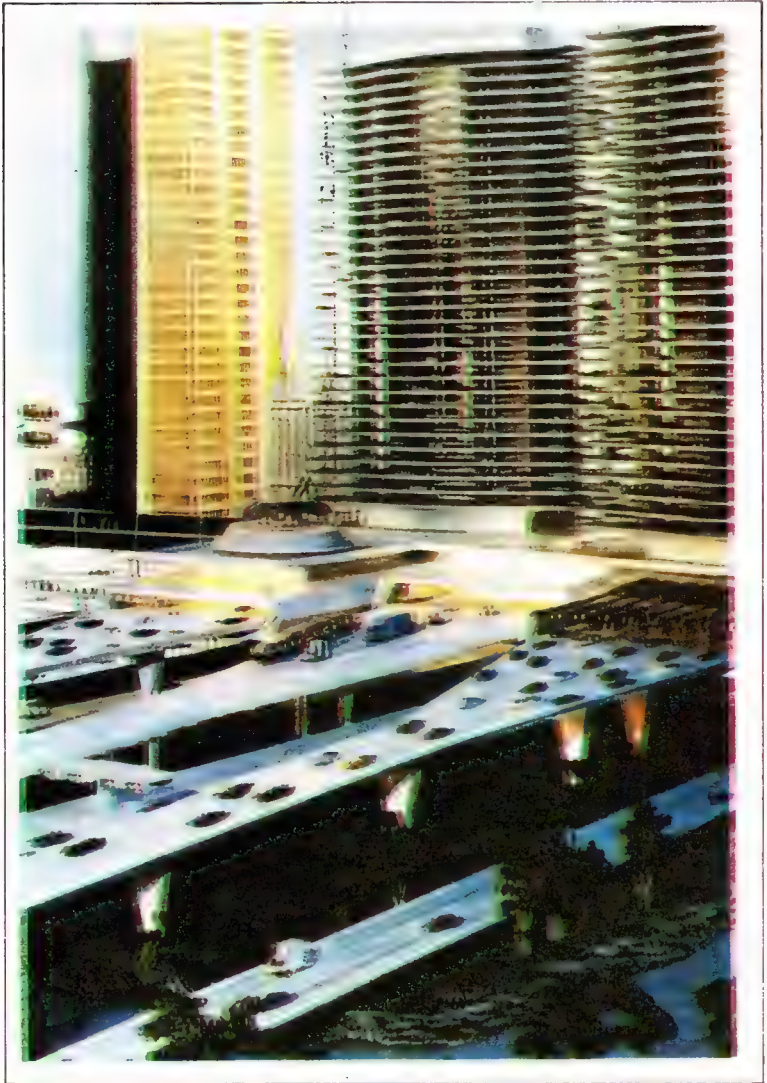
Örneğin:

PRINT TOPLAM

dediğimizde, bizim daha önceden 'TOPLAM' olarak tanımladığımız sonucu bilgisayardan öğrenebiliriz.

Bu komutu kullanmamız için, mutlaka bir sonuç beklememiz

Günümüzün modern kentleri, beraberinde modern bir trafik organizasını da getirmektedir. Bunları sağlıklı bir biçimde çözenin en kısa yolu da bilgisayar aracılığıyla düzenlemelere başvurmaktır. (Karşı sayfada)



gerekmez. Sözelimi, bilgisayara ismini yazdırmak istiyorsunuz. İsmi **Deniz** olan bir arkadaşınızın şu komutu vermesi yeterli olacaktır:

PRINT DENİZ

Bu komutu yazıp gönderdiğiniz an, isminiz çıkacaktır. Bu, isterseniz yazıcıdan da çıkar. Ama ekranı kullanmak her zaman için daha kolaydır. Bazı bilgisayarlarda, yazılacak olan şeyin turnak içine alınması gerekir. Bunun nedenini daha sonra açıklayacağız. Bu tip bilgisayarlarda ise, **PRINT komutu** şu şekilde yazılır:

PRINT 'DENİZ'

sonuç, yine eskisi gibi olacaktır.

PRINT komutu, bir çıkış komutudur. Yani bilgisayardan çıkacak olan şeyle ilgilidir. Bununla birlikte, birçok giriş komutu da vardır. Biraz da bunları açıklayalım.

Bunlardan biri, **LET komutu**dur. Bu komutu da İngilizce anlamıyla birlikte açıklayacağız. Bu kelime İngilizcede **'burakmak, izin vermek, çıkarmak'** anlamlarına gelir. Bilgisayar dilinde de bu tip bir anlama gelir. Yani biz

LET A=4

dediğimizde, bu **'A değişkeninin 4 olduğunu düşün'** anlamına gelir. Bilgisayar, bu değişkenin 4 olmasına izin vermiş olur.

Aynı şekilde, bu komutu şöyle de kullanabiliriz:

LET A=B+C

Bu komutla da, A'nın B ve C'nin toplamına eşit olduğunu belirtiyor, ve bilgisayarın onu bu şekilde kabul etmesi isteniyor.

Şimdilik, bu kadarla yetinelim. Yalnız, hepimizin bildiği gibi, bir program başlama komutuyla açılır ve bitirme komutuyla biter. Bu komutlar da şöyle:

Başlama komutu: START

Bitirme komutu: END

Hepinizin tahmin ettiği gibi, **'start'** kelimesi **'başlamak'**; **'end'** kelimesi de **'bitirmek'** anlamına geliyor.

Şimdi bu öğrendiğimiz komutlarla bir program denemesi yapalım. Bu arada, program komutlarının yanına adreslerini belirten sayıları koymayı unutmayalım. Biz, bu sayıyı 10'dan başlatalım:

```
10 START = 9
20 LET A = 9
30 LET B = 3
40 LET C = 1+B+C
50 LET D = A+B+C
60 PRINT D
70 END
```

Bu program, CPU'ya 9,3 ve 11 sayılarının toplamını almasını ve sonucu yazmasını söylüyor. D değişkeni, programda daha önceden belirtilmiş bir değişken, o yüzden kolaylıkla D'nin yazılmasını söyleyebiliyoruz. Ama eğer D'nin ne olduğunu daha önceden belirtmemiş olsaydık, bilgisayar bize 'D harfini' başardı; ki bu da bizim işimize yaramazdı.

Bu program, üç sayının toplamının çıkarılmasını istiyor demistik, değil mi? Bizim **'değişken'** olarak isimlendirdiğimiz bu harflerle bilgisayar ne yapıyor, anımsıyor musunuz?

Bu değişkenleri veri olarak kabul ediyor ve onların CPU'nun aritmetik bölümüne alıyor. Değişkenleri temsil eden harfler de onların adresleri olmuş oluyor. Yani bilgisayar o değişkeni bulmak isterken, harfine göre arıyor. Bunu daha açık bir şekilde göstereyim:

A	9
B	3
C	11

BASIC, bizim adres yerine harf kullanmamız olanağını sağ-

lar. Bu, bizim için çok yararlıdır. Çünkü, bu harfleri sayılardan daha iyi anımsarız.

Şimdi şöyle bir şey düşünelim. Demin kullandığımız sayılar yerine başka sayılar kullanabilir miyiz? Örneğin 9, 3 ve 11 yerine 2, 14 ve 16'yı toplayabilir miyiz? Doğaldır ki, hayır. Onu yapmamız için, A'yı, B'yi ve C'yi değiştirmemiz gerekir. Ama o zaman da, bu program yalnızca o sayıları toplamaya yarar. Bu programı nasıl değiştirmeden kullanabiliriz? Şu ana kadar öğrendiklerimizle böyle bir program hazırlayamayız.

Böyle bir program için, **INPUT** komutunu kullanmamız gerekir. **Input**, kelime anlamıyla **'giriş, bilgi girişi'** demektir.

Bu programda verileri komutla vermek yerine, komutların kendisinin verileri okumasını sağlayacağız.

Bunu bir örnekle gösterelim:

```
10 INPUT A, B, C
20 LET D = A+B+C
30 PRINT D
40 END
```

INPUT komutu, programı birinin üç tane sayı girmesine kadar durdurur. Sonra verilen sayılar, A, B, ve C adreslerine konulur. D'nin ne için gerektiğini tahmin edebilir misiniz?

Toplamın da saklanacağı ayrı bir adres olması gerekir.

A	9
B	3
C	11
D	23

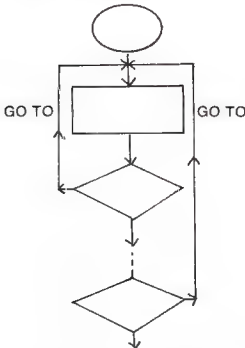
Bu şekilde, sonucu yazması gerektiğinde CPU, onu nerede



bulabileceğini bilir.

Şimdi başka komutları inceleyelim. Akış diyagramlarını incelerken değerlendirme fonksiyonunu anımsıyorsunuz, değil mi? İşte bu fonksiyon için de bir komut vardır.

GO TO komutu, programın bulunduğu adresten başka bir adrese gitmesini sağlar. İngilizce 'go to', 'oraya git' anlamındadır. Bu komut da anlamına uygun olarak, programı komutun içinde belirtilen adrese götürür.



Ama bu komutta bir şey

eksik, değil mi? Biz ne zaman nereye gitmesi gerektiğini söylemedik. Yani bizim değerlendirme sembolümüzde, koşullara göre yön belirleniyordu. Ama biz şu ana kadar öyle bir koşul ileri sürmedik. Bunun için de, İngilizcede 'eğer' anlamına gelen bir komut kullanacağız: **IF komutu**. Bu komut, bize koşulları gösterir.

Son öğrendiklerimizle ilgili bir örnek çözdükten sonra, açıklamalarımızı sürdürelim.

```
10 INPUT A
20 IF A = 0 GO TO 60
30 LET S = S+A
40 LET K = K+1
50 GO TO 10
60 V = S/A
70 PRINT S
80 END
```

CPU, komutlara sırasına göre uyar. Buradaki ilk **GO TO komutu**, eğer A 0'a eşitse programın 'bir sonraki komuta geçmemesini' belirtiyor. Çünkü A'nın 0 olması, programın bitmesi demektir. Biten programı da, son kısma almamız gerekir. Bitmeden önce sonuç çıkarılır, yazılır ve program bitirilir.

İkinci GO TO komutunda, IF koşulu yok. Çünkü bu kez prog-

ramın o adrese gitmesinin koşulu yok. Yani koşul ne olursa olsun, programın geri dönmesi gerekiyor. Çünkü programın bitip bitmediği daha belli değil ve bu ancak programın ikinci adımında belli olacak.

İkinci GO TO komutunda, baştan birkaç komut yineleniyor. Bu, daha önce de gördüğümüz bir tür oluşturuş.

Programı baştan inceleyelim. Bu program, kendisine verilen sayıların ortalamasını alıyor. Biz, burada bir tane **INPUT komutu** verdik. Çünkü sürekli olarak aynı adrese tekrar tekrar kullanacağız. Bütün sayıları aynı anda okuyacağımız yerde, ki bütün bu sayılar için ayrı adresler kullanacağız. Birtek adres saptayıp hepsini teker teker okuyacağız.

Her sayı okunup toplama eklendikten sonra, bizi yeniden **INPUT komutuna** götürür. Böylece program yeni bir sayıyı almaya hazır olur.

Programları bu şekilde yazmak, bize bellekte oldukça fazla yer kazandırır. Biliyorsunuz ki belleğin sürekli olarak boş yere gereksinimi vardır. Biz de programları mümkün olduğu kadar kısaltırsak, belleğin işine yardım etmiş oluruz. Ayrıca prog-

ramızının sonucu daha çabuk çıkacağından, bize de yararı olur.

Belki şu anda programın neden A'nın 0 olup olmadığını soruyorsunuz? Program, bu şekilde sayıların sonuna gelip gelmediğini anlayacaktır. Yalnız bizim de arada 0 girmemesine dikkat etmemiz gerekir. Çünkü arkasından gelen sayılar olduğu halde araya bir 0 sayısı girerse, program kendisini bitirir. Bu şekilde gerçek sonucu almamış oluruz.

Eğer bunu böyle yazmamış olsaydık, program kendisini nerede biteceğini bilemezdi. Yani sonsuza dek kendisine yeni bir sayı verilmesini bekleyebilirdi. Doğallıkla biz böyle bir şeyin olmasını istemediğimiz için, programa ona göre ayarlıyoruz.

Sonra, değerlendirme komutu geliyor. Biraz öncesöylediğimiz gibi, ilk komut koşullu. Eğer A sayısı 0'a eşit olursa program bitiyor. Eğer değilse, 60'taki komuta gidiyor.

10 IF A = 0 GO TO 60

1

60 V = S/A

İkinci GO TO komutu koşulsuz olarak bizi programın başına götürüyor. 30'daki komut, toplama yeni sayıların eklenmesini sağlıyor. 40'taki komutun ne yaptığını söyleyebilecek misiniz?

Bu komut da bizim sayacımız. Her yeni sayı toplama eklendiğinde, o da kendisine bir ekliyor.

Yeni ortalama bulmak istediğimiz zaman toplamı bu sayıya bölüyoruz. Programın 60. komutu da bu işi yapıyor. 70. komut da, sonucun yazılmasını sağlıyor.

Eğer akış diyagramımızı doğru çizmişsek ve ona uygun bir program hazırlamışsak, bunun çalışmaması için hiçbir neden yoktur. Bilgisayar, kaç giriş verdiğimiz verelim, sonucu doğru olarak çıkarır.

Eğer bir programı yanlış yazarsak, bilgisayarın bunu bulması gerekir. Her insan, program yazarken bazı yanlışlar yapabilir. Yalnız bu yanlışlar ortadan kaldırılmazsa sonuçta çok yanlış şeylerin çıkmasına neden olabilir.

Bu yanlışların ortadan kaldırılması işlemine ayıklama dendiğini söylemiştik. Ayıklama işinin bir kısmını programın kendisi yapar. Çevirici, programı iki-

lik düzene çevirirken, dikkatle komutları inceler. Eğer anlamadığı bir şey rastlarsa, bunu hemen belirtir.

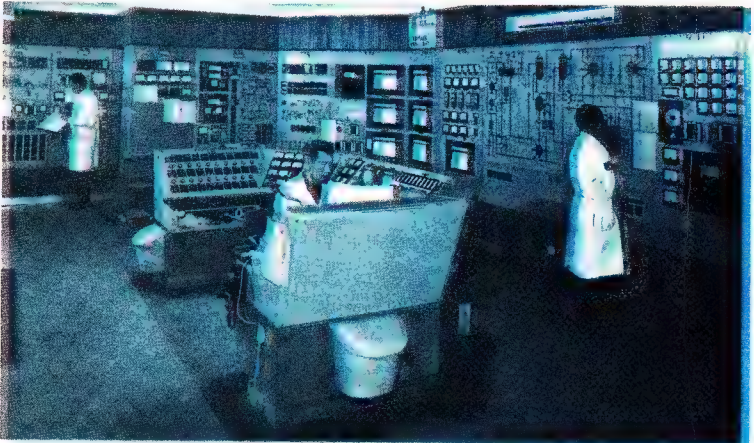
Örneğin komutlardan birini yanlış yazarsanız, bunu hemen çıkarabilirsiniz. Çünkü çevirici bunu hemen anlayacaktır ve hemen size bildirecektir.

Ancak bazı ayıklamaların programcı tarafından yapılması gerekir. Program ne kadar karmaşık olursa, bu ayıklama işi de o kadar karışık ve zor olur.

Yalnız son zamanlarda programları çevirdikten sonra da kontrol eden programlar yapılmıştır. Bu programlar yardımıyla, programcının ayıklama işi de olabildiğince basitleştirilmiştir. Bu tip programlara 'sistem programı' adı verilir.

Sistem programları, bilgisayarın belleğinde yer alır. Programın çıkarılmasına yardımcı olurlar.

Örneğin monitör de bir sistem programıdır. ROM içinde yer alır. Bu program CPU'nun içinde giren çıkan programları kontrol eder, onların doğru sırada olmasını sağlar. Bu da bir bakıma ayıklama işidir.





Ama bir monitörün bundan daha fazla görevi vardır. Çok kalabalık, karışık bir kütüphanede çalışan bir kütüphaneciye benzerler. Eğer bilgisayarı bir kütüphane olarak düşünersek, onun kitapları programlarıdır. Yeni bir program yazıldığında, bu kütüphaneye yeni bir kitap eklenmiş demektir. Kütüphaneci bu kitabı alır ve uygun raflardan birine koyar. Monitör de programı yan belleklerden birinin içine yerleştirir.

Programın çıkarılması, kütüphaneden bir kitabın ödünç alınmasına benzer. Siz istediğiniz kitabı kütüphaneciye söylersiniz, o da size istediğiniz kitabı arayıp bulur, verir. Bilgisayarda da, siz bu kez aradığınız programı istersiniz. Monitör de, size bu programı yan belleklerden bulur, çıkarır ve verir.

Sonra o kitabın ödünç verme masasına götürülmesi gerekir. Aynı şekilde program da belleğe kopya edilir.

Ödünç verme masasında, kitabınız damgalanır. Damgada

hangi tarihte getireceğiniz basılıdır. Monitör de çeviriciyi çağırır ve ikilik düzene çevrilmiş programı CPU'ya geçirir.

Kitabı bitirdiğinizde, onu yeniden kütüphaneye götürür, bir yenisini alırsınız. Kütüphaneci sizden eski kitabı alır, yerine koyar ve size istediğiniz kitabı getirir. Bunun gibi, monitör de eski programı alır, çıkarır, yenne yeni programı koyar.

Herhangi bir sorunuz olduğunda, kütüphaneciye sorarsınız. O size yardımcı olur. Burada da yine bu işi monitör görür. Bilgisayarların kullandığı yerlerde, insanların sürekli terminalleri bir şeyler yazdığını görürüz. Bu insanlar;

bilgisayara veri girişi yapıyorlardır.

bilgisayardan veri istiyorlardır.

program yazıyorlardır ya da program çıkarımı yapıyorlardır.

Bütün bu işler monitör tarafından yapılır.

Monitörün de bir program olduğunu unutmayalım. Bu prog-

ramı çok deneyimli programcılar yazabilir. Bildiğiniz gibi bilgisayarlar, programsız hiçbir şey yapamazlar. Programları da insanlar yazabildiğine göre, bilgisayarlar insanlar olmadan hiçbir işe yaramazlar demektir.

Bilgisayarlar, insan yaşamını kolaylaştırmak için gerçekleştirilmiş bir araçtır. İnsan olmadan, kendi başına hiçbir şey yapamaz.

Donanım ve Kullanım Bir Arada

BİLGISAYARLARIN insan yaşamını kolaylaştırmak için gerçekleştirildiğini biraz önce belirttik. Donanım, bilgisayarın makine kısmıdır. Yani donanımın içine CPU, disket, teyp, yazıcı ve çıkış terminalleri girer. Donanım, bilgisayarın teknik kısmıyla uğraşır. Kullanım ise, bilgisayar donanımının çalışması için gereken yazılı bilgilerdir.



Birçok sistemlerde, donanım ve kullanım bir arada çalıştırılır. Örnek olarak hastaneleri, fabrikaları gösterebiliriz.

Sizin bir mühendis olduğunuzu var sayalım. Biri, sizin ev için bilgisayar kontrollü bir ısıtma sistemi kurmanızı istiyor. Bu ısıtma sistemi güneş enerjisiyle çalışacak.

Bu ısıtma sisteminin nasıl çalışacağını düşününce ne olur mu? Evin çatısına, toplayıcı metal tabakalar konur. Bu tabakalar, güneşli havalarda ısınır. Bodrum katında da bir su deposu bulunur.

Güneşli havalarda, ısınan metal tabakalar, alt taraftaki suya ısı gönderir. Bu ısı orada depolanır. Soğuk havalarda da, tıpkı kalorifer sistemi gibi bu sıcak suyla ev ısıtılır.

Güneşin olmadığı zamanlarda, yani gecelevin, bulutlu ve soğuk havalarda tabakaları ısı enerjisi toplayamaz. Bu nedenle otomatikman kendini kapatır.

Eğer bu kapalı günler çok fazla arka arkaya gelirse, depodaki enerjinin tümesi kullanılır. Bu yüzden yalnızca bu sisteme güvenmek olmaz. Yedekte gaz, elektrik ya da kömürle çalışan bir ısıtma sistemi de olmalıdır. Bu sistem, depoda enerji kalmadıkça zaman devreye girer ve ısıtma ıstır.

Şimdi, birçok evlerde çok modern bir sistem kullaılmaktadır. Evler, güneş enerjisi ile ısıtılır ve bu yedek sistemi de "termostat" denilen bir cihaz kontrol eder. Termostatın amacı, evin ısıtılmasını biliyor musunuz?

Termostat, bir dedektör sıcaklığı ayarlar. Eğer sıcaklık bu derecenin altına düşerse, termostat ısıyı yükseltir. Eğer çok fazla ısındıysa, dereceyi düşürür.

Bu işi yapmaması için termostatın ısıya karşı duyarlı bir parçası olması gerekir. Bu parçalar, bilgisayarların kimyasal soldüsyonların sıcaklığını



ölçmek için kullanılan parçalara sahiptir.

Bu termostatın gerçek verilerine yerleştirilen bir parçalar yardımıyla, termostatın her verindeki sıcaklığı algılayabilir. Eğer bu sıcaklık düşerse, bilgisayar, termostatın ısıtılmasını alır.

Termostatın duyarlı parçaları, termostatı kullanır. Ama bu termostat, sıcaklığı bilgisayarla birleştirir. Bilgisayar da termostattan çok daha karmaşık bir yapıya sahip olduğuna göre, bir de soldüsyonlarda kullanıldığı gibi birçok duyarlı parçaya kullanılır. Bu parçalar da, evin ısıtılmasını sağlar. Bu parçaların, doğal olarak odalar

Evin ortalama sıcaklığı istenilen altına düşerse, bilgisayar hemen sıcaklığı artırır. Eğer çok sıcak olursa, sistemi kapatır. Bu şekilde sıcaklık düşer.

Bütün bunlar, bilgisayarın donanımıyla ilgilidir. Ve bu bilgisayarın kendisini, klavyeyi ve bilgisayarın saklanacağı bir bellek ünitesini kapsar. Ayrıca sıcaklık ölçen duyarlı parçalar ve bütün evin sıcaklığını ölçebilen bütün bir sistem vardır.

Bizim bir de kullanımı bilmemiz gerekir, değil mi? Bütün bu sistemi kurduktan sonra, kullanmasını bilmeysek hiçbir işe yaramaz. Bilgisayarları, bir insan tarafından kullanılmadığı zaman kendilerini çalıştırır.

Bu sistem, bir program yazmamız gere-

kir. Bunun için de akış diyagramını çizmeliyiz.

Başlangıç komutundan sonra ilk komutumuz, girişi okutacak bir komut olmalı. Girişimiz, du - yarlı parçalar tarafından okunacak olan sıcaklıklar ve depodaki suyun sıcaklığıdır. Çünkü depodaki sıcaklık evi ısıtmaya yetme - yecek kadar azsa, yedek sistemin devreye girmesi gerekir.

Ayrıca bilgisayara evde iste - diğimiz sıcaklığı da bildirmemiz gerekir. Bunun da girişte veril - mesi gerekir. Çünkü sonradan bu sıcaklığı değiştirmek isteyebiliriz. Bunun için de bütün programı değiştirmeden kullanabileceği - miz şekilde yazmamız gerekir. Ancak bu sıcaklığı hiçbir zaman değiştirmeyi düşünmüyorsak, programın içine koyabiliriz:

LET V = 20

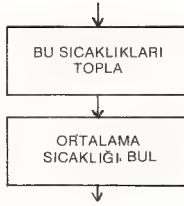
Ama sonradan değiştirebile - ceğimiz sayıları, kesinlikle girişte vermemiz gerekir. Ancak bu şekilde bilgisayar değişen sayıları gerektiği gibi kullanabilir.

INPUT V

Böylece, akış diyagramımızı şu şekilde yazabiliriz:



Girişi yazdıktan sonra, du - yarlı parçaların okuduğu derece - leri vermemiz gerekir. Bilgisayar bunların ortalamasını alır.



Sonra da değerlendirme komutuna sıra geliyor. Evin sıcaklığının istenilen derecenin altında mı üstünde mi olduğunu belirtmemiz gerekiyor. Bu yüz - den önce bir değerlendirme komutu koyacağız:



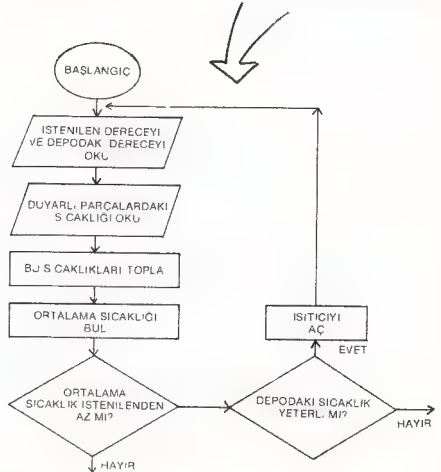
Eğer sıcaklık istenilenin altın - daysa, sıcaklığı arttırmamız gere - kir. Sistemi açacağız. Ama önce depodaki sıcaklığın bizim isteği - mize yetecek kadar olup - olmadığını öğrenmemiz gereki - yor.



Depoda yeterince sıcaklık olduğunu varsayalım. Sıcaklığın evin içinde istediğimiz düzeye geldiğini nasıl anlayacağız?

Bunun için evin içindeki sıcaklığı sürekli okumamız gere - kiyor. Aynı zamanda depodaki sıcaklığı da sürekli ölçmemiz gerekir. Çünkü evi ısıttıkça depo - daki sıcaklık da düşer.

Bu arada girişte verdiğimiz verilerin de değişmediğinden emin olmamız gerekir. Yani bir de girişi okumamız gerekir. Bu ise, bir tur yapmamız demektir.





Dünyanın en çok satan bilgisayarlarından biri olan, "Apple II"nin iç düzeni. (Üstte)

Güzel, şimdi şunu düşünelim. Eğer depodaki suyun sıcaklığı evin sıcaklığından daha az olursa, ne olur?

Bu, depodaki sıcaklığın evi ısıtacak kadar fazla olmadığını gösterir. Depodaki sıcaklık yeterli olmayınca, yedek sisteme başvurulur. Bu sistem, daha önce

belirttiğimiz gibi gaz, kömür, vs. olabilir.

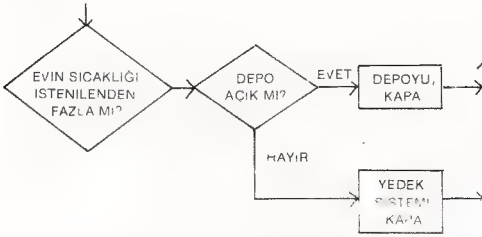
Bundan sonra yeniden başa dönüp, her adımı bir kez daha yapmamız gerekir. Ancak bu şekilde ısıtıcının ne zaman kapatılacağı konusunda bir fikrimiz olur. Bunu da evin ısısının istenen dereceye ulaşmasından anlayacağız. Bu bizim ilk değerlendirme komutumuzdu, değil mi?

Şimdi yeniden başa dönelim ve evin sıcaklığının istenen sıcaklıktan fazla olması halinde ne yapılıyor, onu inceleyelim.

Evin istenenden sıcak olması, ısıtıcının gereğinden çok çalıştığını gösterir. Bu durumda ısıtıcıyı kapatmamız gerekir.



Ama hangi sistemi kapatacağız? Asıl sistemi mi, yoksa yedek sistemi mi? Hangisi çalışıyorsa diyeceksiniz. Ama biz bunlardan hangisinin çalışmakta olduğunu nasıl anlayacağız? Bunun için yeni bir değerlendirme komutu koymamız gerekiyor

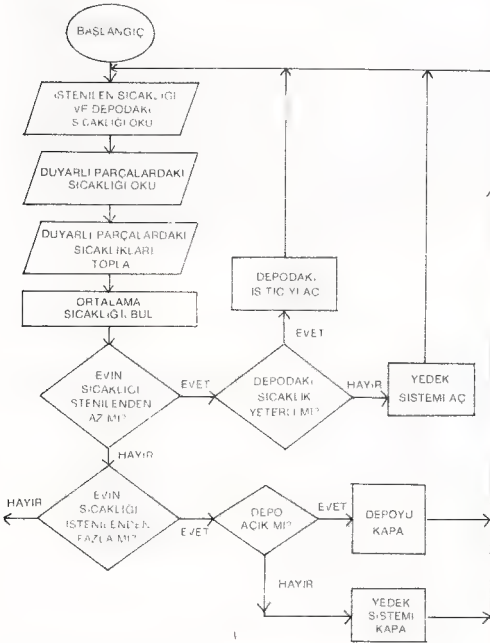


Bu son komutumuz, bize hangi sistemi kapatmamız gerektiğini gösterecek. Doğaldır ki,

bundan sonra sıcaklıkları yeniden kontrol etmek için bir tur daha koyacağız.

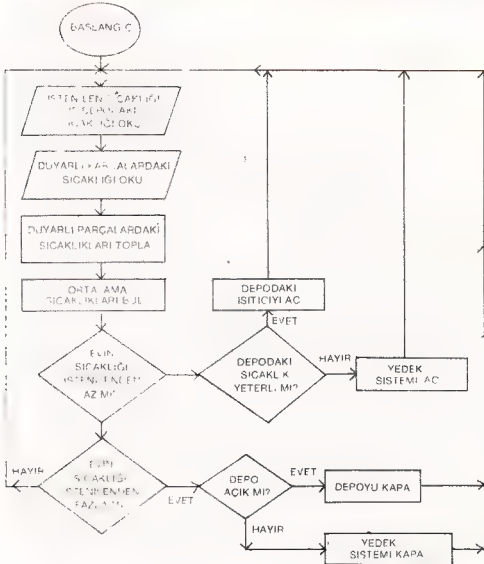
yacağız. Ama bu bizim akış diyagramımızı orada bitirmemiz anlamına gelmez, tekrar tekrar sıcaklıkları kontrol etmemiz gerekir. Hatta deponun sıcaklığını bile. Çünkü eğer herhangi biri hastalanmış olabilir, ya da sıcaklık fazla gelmiş olabilir. Bu nedenle istenilen sıcaklık oranı değiştirilmeside olabilir. Bilgisayarın bunu da izlemesi gerekir.

Akış diyagramının, artık bitmek üzere. İsterseniz şimdiye kadar yaptıklarımıza bir göz atalım. Akış diyagramımız, sürekli odaların sıcaklığını ölçerken bir vandan da bu sıcaklığı istenilen sıcaklıkla karşılaştırıp gerekirse değiştiriyor, ayrıca depodaki sıcaklığı da kontrol ediyor. Depodaki sıcaklık yetersizse, yedek sistemi devreye sokuyor. Sıcaklık fazla gelirse de sist. kullanyırsanız kapatıyor. Eğer sıcaklık istenilen düzeydeyse, bu kez sistemi kapatıyor, ama kontrol etmeyi de bırakmıyor. Bütün bunları içeren akış diyagramımızı çizelim:





"Apple II" sistemi (Ustte)



Şimdi sıra programımızı yazmaya geldi. Önce kullanacağımız dili seçmemiz gerekiyor. Biz BASIC öğrendiğimiz için, programı BASIC'le yazacağız.

Bütün programı yazmamız gerekmiyor. Bu çok uzun ve karışık bir program olur. Onun yerine, bazı komutları göstereceğiz. Bu komutlar, programdaki belli başlı işleri yapacak olan komutlar olacak.

Önce, giriş ve çıkış değerleri için bazı adresler göstermemiz gerekiyor. Daha önce kullanmak istediğimiz değerlere A, B gibi adresler verdiğimiz gibi, bu değerlere de adresler vermemiz gerekiyor.

İlk girişimiz, evde istenilen sıcaklık olacak. Bu değere iş adresini verebiliriz. Daha önceki değerlerin adresleri hep tek harfli oluyordu, şimdi niye böyle iki harfli bir adres verdiğimiz merak edebilirsiniz. Adresler, birden fazla harften meydana gelebilirler. Doğaldır ki, bunların da bir sınırı vardır. Genelde alt karakter ve sayıya izin vardır. Sayı dedik, bu adreslerde sayılarda

Özetikle turizm şirketleri tarafından kullanılan, "Commodore 8082" bilgisayar. (Sağda)

kullanılabilir. Yalnız bu sayılar, adreslerin başına gelemez. Ancak ortasına ya da sonuna gelebilirler. Bu gibi adreslere birer örnek verelim:

A PROG R6 SAYI ALTI D3M7

Biz, kendi örneğimiz için iki karakterlik bir adres kullandık. Şimdi ikinci girişimizi yazalım. Bu, deponun sıcaklığıydı. Onu da DS ile gösterebiliriz.

Bir de duyarlı parçalarımız var. Onlar da odalardaki sıcaklıkları ölçüyor. İş fazla karıştırmamak için, evin altı odası olduğunu düşünelim. Yani bizim altı tane duyarlı parçamız sıcaklığı ölçecek. Bunların hepsine birden DP diyelim.

Altısının birden aynı adresi kullanması, bunların aynı adreste saklanacağı anlamına gelmez. Bunlardan yalnızca biricisi bu adreste saklanır. Diğerleri de, bu adresi izleyen adreslerde saklanırlar. Bunu bilgisayarın anlaması için, adreslerin sonuna birer ek koyacağız. Bu ekler, belli bir grup kullanıldığında zaman işe yararlar. Bütün gruba aynı isimler verilir. Buna karşılık ekler de, grubun kaçmıçı elemanından söz ettiğimizi gösterir. Bütün bu anlattıklarımıza göre, bu duyarlı parçaları şu şekilde yazabiliriz:

DP (1) DP (2) DP (3) DP (4)

Bu ekleri parantez içine alıp yazabiliriz. Şimdi de sonuc için bir adres belirtmemiz gerekiyor. Ne tür sonuçlarımız, çıkışlarımız var, onları inceleyelim. Programın depoyu kapatması ya da açması gerekebilir. Ayrıca bir de yedek sistemimiz var. Yani bizim elimizde bir depodaki istisya ile ilgili bir çıkış olacak, bir de yedek sistemle ilgili bir çıkış olacak.

Depodaki istisyaçı DI olarak gösterebiliriz. Yedek sistem içinde

YS harflerini kullanırız. Bunların alabileceği değerleri tahmin edebiliyor musunuz?

Bunlar 0 ve 1'dir. Gayet kolay, değil mi? Çünkü bu sistem ya açık, ya da kapalı olacak. Evet, başka adreslere gereksinimiz var mı?

Bir tane duyarlı parçaların okuduğu değerlerin toplamı için, bir tane de bu değerlerin ortalamasını almak için iki tane adrese gereksinimiz var. Bunlara da DT ve DC diyelim. Şimdi de komutlara bir göz atalım

Bir programda, komutların gereken sıra içinde yazılması gerekir. Tıpkı daha önceden gösterdiğimiz yemek tarifinde olduğu gibi. Ancak biz bu komutların ne yaptığını tam olarak kavramak için akış diyagramının yanına yazacağız. Orada yaptıkları işlerin ne anlama geldiğini daha kolay, daha rahat kavrayız.

Programda INPUT komutu yerine PORT komutu kullanılmış. Çünkü genelde INPUT komutu giriş terminallerinden

verilecek veriler için kullanılır.

Oysa bizim programımızda, istenilen sıcaklık (İS) duvarındaki bir düğme üzerinden okunacak. Bu düğmeyi, termostata benzetebiliriz. Deponun sıcaklığı (DS) da, odalarda bulunan duyarlı parçalara benzer bir aletle ölçülüp bilgisayara gönderilir. Bu alet, deponun sıcaklığını sürekli olarak ölçüp, değişimleri haber verir.

Böylece ilk komutlarda İS ve DS'yi bilgisayara vermiş - okutmuş - olduk. Burada kullanacağımız bir başka yabancı komut daha var. Bu komut, genelde başka bir komutla birlikte kullanılır ve ikisi birlikte sayacı görevini görür.

Bunlardan biri, FOR=TO komutudur. Bu komutta, FOR'dan sonra, kullanılacak dizinin ilk rakamı yazılır. TO'dan sonra da dizinin son rakamı yazılır. Buna bir örnek verelim. İki den sekize kadar olan sayıların karelerini alacağımızı düşünelim. Bunun için FOR = 2 TO 8 deriz. Bu şekilde bilgisayar, bizim 2'den 8'e kadar olan sayıları kullanmak istediğimizi anlar.





10 PORT IS
20 PORT DS

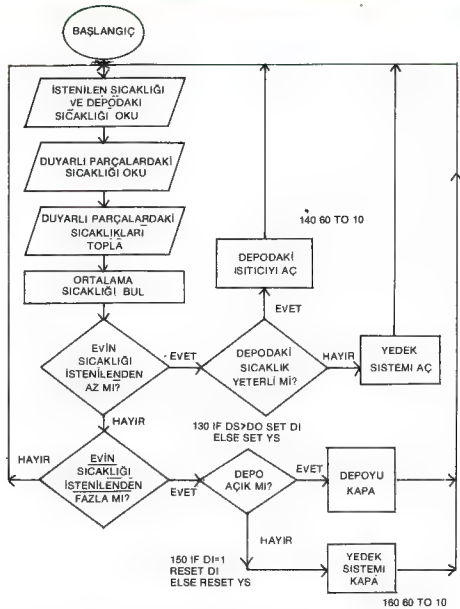
30 FOR I = 1 TO 6
40 PORT DP (I)
50 NEXT I

60 FOR I = 1 TO 6
70 DT = DT + DP (I)
80 NEXT I

90 DO = DT/6

100 IF DO<5
60 TO 130

110 IF DO>5
60 TO 150
120 60 TO 10



Bu komutla birlikte kullanılan diğer komut da NEXT I komutudur. "Next", İngilizcede "bir sonraki" anlamına gelir. Bu komutla birlikte, elimizdeki sayı yerini bir sonrakine bırakır.

Bunu daha açık bir şekilde gösterebiliriz. Deminki örneği daha detaylı olarak incelersek:

FOR I = 2 TO 8

NEXT V I

Yukarıdaki örnekte ikiden sekize kadar olan sayıların, sırayla herhangi bir işlemde kullanılması sağlanıyor. Örneğin bu sayıların karesi alınabildiği gibi, birbirleriyle toplanabilir, ya da başka bir şey yapılabilir.

Bizim programımızda, 30. ve 50. komutlar arasında, duyarlı parçalar (DP) tek tek okunuyor. Buradaki I, sayaç görevini üstlenmiştir. Sayaç yardımıyla birinci parçadan altıncı parçaya kadar hepsi sırayla 30. ve 50. komutlar arasını geçerler. Bu da bir tur meydana getirir.

60. ve 80. komutlar arasında da bir tur vardır. Bu turda da komutlar, her DP'yi toplama (DT) ekler.

90. komut, değer ortalamasını (DO) çıkarır. Anımsarsanız, odanın ortalama sıcaklığını bulmak için odalardaki sıcaklıkların ortalamasını alıyorduk.

100. komut, eğer DO, İS'den küçükse, akış yönünü 130. komuta doğru değiştiriyor.

Eğer DO, İS'den küçük değilse, onu bir sonraki komuta gönderiyor. Bu komutta da eğer DO, İS'den büyükse, akış yönü 150. komuta doğru değişiyor.

Bu iki komut, ısıtıcının açılıp kapanmasını ayarlıyor. Yani eğer evin sıcaklığı istenen sıcaklıktan küçükse ısıtıcı açılıyor; büyükse ısıtıcı kapanıyor.

Ama, DO, İS'den ne büyük ne de küçük olmayabilir. Eğer even sıcaklığı itenen sıcaklığa eşitse, bilgisayar evin sıcaklığını ölçmeyi sürdürüyor, başka bir değişiklik yapmıyor. Programda da eğer ikisi birbirine eşitse, 120. komuta geçiyor.

Şimdi programın ısıtıcıyı çalıştıran bölümünü inceleyelim. 130. komut, eğer DS, DO'dan büyükse; DI'yi 1'e getirilmesini, yani açılmasını söylüyor. Bu komut, programda SET ile gösterilmiş. "Set", arkasından gelen adresi 1'e getirme işini görüyor. Yani "SET DI" dediğimizde bu, DI'yi 1 konumuna getir, aç demek oluyor.

Aynı komut içinde, bir de "ELSE" sözcüğü var. Bu, "eğer öyle değilse" anlamına gelmektedir. Yani eğer DS, DO'dan büyük değilse anlamına gelmektedir. Bu koşulda, bilgisayar "SET DI" yapmayacak; onun yerine "SET YS" yapacak. Başka bir deyişle YS'yi 1 konumuna getirecek, açacak.

Şimdi de 140. komuta bakalım. Bu komut, yeni bir değer okumak için başa dönmemizi öngörüyor.

Artık programın sonuna geldik. 150. komut, eğer DI 1 ise (açıksa), onu 0 konumuna getirir, yani kapa diyor. Burada yeni gördüğümüz bir komut daha var. "RESET" komutu.

"SET" komutunu anımsayalım. Bu, belirtilen adresi 1 konumuna getir demek oluyordu. "RESET" komutu da, belirtilen adresteki bilgiyi 0 konumuna getir anlamına geliyor. Yani "RESET DI", yedek sistemi 0 konumuna getir, kapa anlamında kullanılıyor.

Burada bir de "ELSE" komutu var. Bu bildiğiniz gibi "eğer öyle değilse" anlamına gelmekteydi. Bu komutta da eğer DI açık değilse YS'nin açık olmasını

gerekir, yani onu kapayacağız diyor.

160. komut, bir başka değeri almak için, programın akış yönünü başlangıca doğru yönlendiriyor.

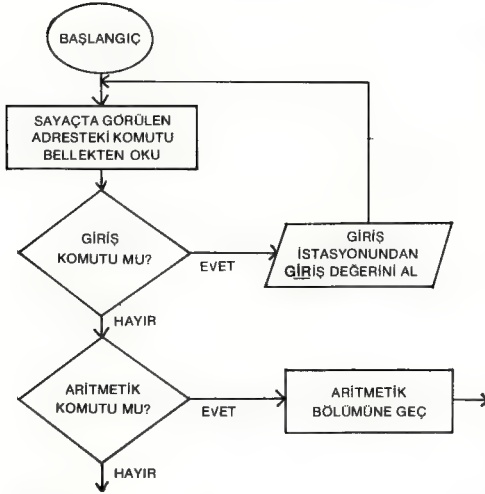
Bu böylece sürüp gidiyor.

Bir program, bu şekilde hemen çalışmaya başlayamaz. Programımız tamam, ama bir ismi yok. Bir programı çalışır hale getirebilmek için, ona önce bir isim vermemiz gerekir. Biz, programımıza bir isim vermezsek, çevirici onu bulamaz. Bizim o programı yeniden isteyebilmemiz için ona bir isim vermemiz gerekir. Biz bu programa ISITMA ismini verelim.

Biz bu ismi yazar yazmaz, çevirici onu hemen belleklerden birine götürür, saklar. Bu arada da ikilik düzene göre programı yeniden yazar. Aslında bu program daha bir yazmaya başlar başlamaz, ikilik düzene çevilmeye başlanmıştır. Bu şekilde çevrilen program, doğrudan ana belleğe yazılır.

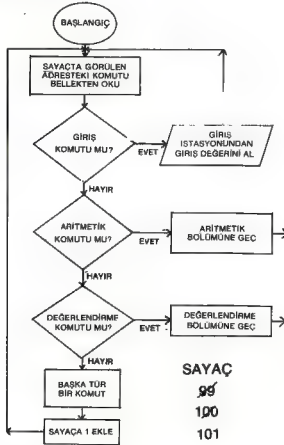
Sonra, programın bilgisayarından çıkartılması işlemi yapılır. Yani bilgisayar programı yeniden alır, çalıştırır; ve biz de bu arada onu kullanırız. Bu çıkartım işi için de özel bir komut kullanılır. Bu komut, "RUN" komutudur. Biz, "RUN ISITMA" dediğimiz zaman bu program çağrılır ve kullanılacak durumda bekler.

Artık bilgisayarın kullanım kısmı devreye girmiştir. Önce kontrol bölümü programı bellekten alır, komutları tek tek inceler. Öngördükleri işlemleri yapar. Komut, bir giriş okumasını söylüyorsa, başa gidip onu okur. Bu okuma işlemi sırasında, bilgi bellekten taşıt aracılığıyla gelir. Eğer komut bazı sayıların toplanmasını ya da birbirine bölünmesini söylüyorsa, bu işi aritmetik bölümü yapar; ve sonucu belleğe geri gönderir.



Ve eğer deminki örnekte olduğu gibi bir değerlendirme komutu ile karşılaşırsak, bu işi de mantık bölümü gerçekleştirecek.

Programdaki komutların CPU' da sırayla yapılmasını sağlayan sayacıdır.



Gördüğünüz gibi bu programdaki girişler ya duyarlı parçalardan, ya da duvardaki dereceden öğrenilir. Ama bilgisayara giriş vermenin türlü şekilleri vardır. Örneğin bir giriş terminalinin klavyesinden bu iş rahatlıkla yapılabilir. Bu gibi durumlarda, kullanıcı bilgiye girmeden önce, ekranda daha önceden belirlenmiş bir mesaj ortaya çıkar. Bu mesaj yardımıyla doğru zamanda bilgi girişi yapılmış olur.

Bunu yapmak için, programa bir de **PRINT** komutunu eklememiz gerekir. **PRINT** komutunun ne yaptığını anımsıyorsanız, değil mi? Herhangi bir şeyin yazılı olarak çıkması için bu komutu kullandık. Burada da şu şekilde kullanacağız:

10 PRINT "SAYINI YAZ"
20 INPUT ID

Bu tipte hazırlanmış bir programda, yanlış giriş yapma olasılığı çok azdır. Çünkü giriş hakkında ufak ipuçları verilebilir. Özellikle çok girişli programlarda, kullanıcıya büyük kolaylık sağlar. Bu mesaj belirlediği andan itibaren, giriş yapıldıkça program durur ve o bilgiyi almadan da işi götürmeye başlamaz.

Örneğin hastanelerde, hasta kartlarının programları bu şekilde hazırlanabilir:

HASTANIN İSMİ:
KAN GRUBU:
TANŞİYONU:
BELİRTİLER-OLANLARIN
YANINA "E" TUŞUNA BA-
SIN:

İŞTİHAZSIZLIK
SÜREKLİ BAŞ AĞRISI
BURUN AKMASI

vs.

PRINT komutu, bütün bu mesajların ekrana çıkmasını ve yanıtı gelene kadar beklenmesini sağlar.

Çalınan arabalar hakkında rapor tutan bir polis bürosunun da programında şunlar yazılı olabilir:

ARABANIN YAPIM TARİ-
Hİ:

"Apple II" mikrobilgisayırıyla
çalışan bir öğrenci. (Sağda)

MODELİ:

RENGİ:

İHBAR TARİHİ:

SAHİBİNİN İSMİ:

Bu mesajlara göre görevli de,
doğru bilgileri kayda geçirebilir.

Bu komut, en çok öğretmen-
likte işe yarar. Doğru yazmayı,
işlem yapmayı öğreten birçok
program vardır. Ayrıca yetişkin-
lere de özel birtakım şeyler öğre-
ten programlar vardır.

Hatta insanlara nasıl prog-
ram yazılacağını öğreten prog-
ramlar da vardır.

Aritmetik öğreten bir prog-
ram, şu şekilde olabilir. Zeynep
isminde bir arkadaşınız, bilgi-
sayarın sorduğu soruları yanıtlı-
yor:

İSMİNİ YAZ: Zeynep

ZEYNEP, 2+2 KAÇ EDER 4

ÇOK GÜZEL PEKİ, 3+7

KAÇ EDER 11

PEK DEĞİL. BİR DAHA

DENEMEK İSTER MİSİN? 10

BU İYİ İŞTE, PEKİ 5—2

KAÇ OLUR? 3

ÇOK GÜZEL. 3 SORUYU

DOĞRU YANITLADIN, ZEY-

NEP:

Bilgisayarın, verilen yanıtın



doğru olup olmadığını nasıl çıka-
rıldığını merak ediyorsunuz-
dur.

Bunun bir şekli, bilgisayarın
aynı işlemi kendisinin de yap-
masıdır. Sonra kendisininkiyle
Zeynep'in yanıtını kıyaslar.

Başka bir şekil ise, doğru
yanıtları bir liste halinde bellekte
saklamaktır. Böylece öğrenci her
yanıt verişte, bu yanıt listedekiyle
kıyaslanır ve değerlendirmesi ona
göre yapılır.



Bu mesajlar, soruları soran
programın bir parçasıdır. Onlar
da bellekte saklanmaktadır.

Öğrencinin ismi de, verilerin
arasındadır. Diğer girişlerle bir-
likte onun da girişi yapılır. Bilgi-
sayar, bunu belleğin yine ayrı bir
yerinde saklar ve gerektiğinde
mesajın içinde kullanır.

CPU'nun doğru yanıtların
sırasını tutan bir sayacı kullan-
ması gerekir. Bu şekilde, doğru
yanıtı da ekranda gösterebilir.

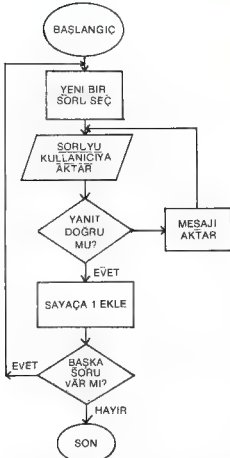
Böyle bir programın akış
diagramını şu şekilde olabilir:

PROBLEM NÖ	ÖĞRENCİNİN YANITI	DOĞRU YANIT CETVELİ								
		<table><tr><td>1</td><td>4</td></tr><tr><td>2</td><td>10</td></tr><tr><td>3</td><td>3</td></tr><tr><td></td><td>⋮</td></tr></table>	1	4	2	10	3	3		⋮
1	4									
2	10									
3	3									
	⋮									
2	<table><tr><td>10</td><td>AYNI MI?</td></tr></table>	10	AYNI MI?							
10	AYNI MI?									

Bu işi de mantık bölümü
yapar. Çünkü içinde değeren-
dirme koşulları vardır. Eğer
öğrencinin yanıtıyla doğru yanıt-
lar uyarsa, ekranda şöyle bir yazı
belirir:

ÇOK GÜZEL

Eğer yanıtlar birbirini tut-
mazsa, öğrenciye bir daha dene-
mesini söyleyen bir yazı belirir.
Doğaldır ki bu yazıları bilgisayar
kendiliğinden yazamaz. Bilgi-
sayar kendi başına hiçbir şey
yapamaz. Yalnızca kendisine
söylenilene yapabilir.



Bir program, doğru yazmayı da öğretebilir demiştik. Bunu nasıl yapabileceğini düşünabiliyor musunuz? Bir kelimenin doğru yazılışını, ayrıca öğrencilerin doğru yazıp yazmadığını nasıl bilebilir?

Bu tip programlar, menü gibi hazırlanır. Yani birçok seçenek verilir, doğru yanıtın bulunması istenir. Size bir örnek verelim:

ASAĞIDAKİ SÖZCÜKLERİ DİKKATLE İNCELEYİN

1. SOKAKIM
2. SOKAKİM
3. SOKAĞIM

DOĞRU YAZILAN SÖZCÜĞÜN NUMARASINI YAZIN

HAYIR, BU DOĞRU DEĞİL.

ŞU KURALI ANIMSA:

SONU K İLE BİTEN SÖZCÜKLER SESLİ EK ALDIĞINDA K DEĞİŞİR.

ŞİMDİ, YENİDEN DENE:

1. YALNIŞ
 2. YANLIŞ
 3. YANLIZ
- vs.

Bu tip programlarda, doğru verilmeyen bir yanıt karşısında dilbilgisinin bin kuralı anımsatılıyor. Ayrıca çok yanlış yapılan bazı sözcükler kapsama alınmış. Bu tip bir programla, öğrencilerin yazma yanlışlıkları en aza indirilebilir.

Sürekli aynı tip yanlışlar yapan bir öğrenciye, zayıf noktasını geliştirecek örnekler verilir. Yanlışın tipine göre, programın akış yönü değişir.

Bütün bu örneklerin, mesajların, seçeneklerin, bellekte yer alması gerekir. Ama, bunlar çok yer tutarlar. O yüzden çoğunlukla disklerde, disketlerde saklanırlar. Bazen bu değişik bilgiler, disklerde değişik dosyalarda saklanırlar. Dosyalar, büyük kapsamlı bilgi gruplarıdır. Anımsıyorsunuz, değil mi?

Bu örnekler, mesajlar, vs. bir ya da birkaç dosyanın içine sığdırılmış olabilirler. Sonra da CPU gerek duyduğu kısımları ana belleğe kopya eder, oradan kullanır.

Kayıtların ne olduğunu da anımsarsınız herhalde. Bunlar, dosyaları oluşturan parçalarıdır. Bir sayfadaki cümlelere benzetebiliriz onları. Bizim programımızdaki mesajların her biri de birer kayıttır:

EVET, DOĞRU YANIT

**HAYIR, YANLIŞ YANIT
BİR DAHA DENE
ÇOK İYİ**

**...TANE DOĞRU YANIT
VERDİNİZ**

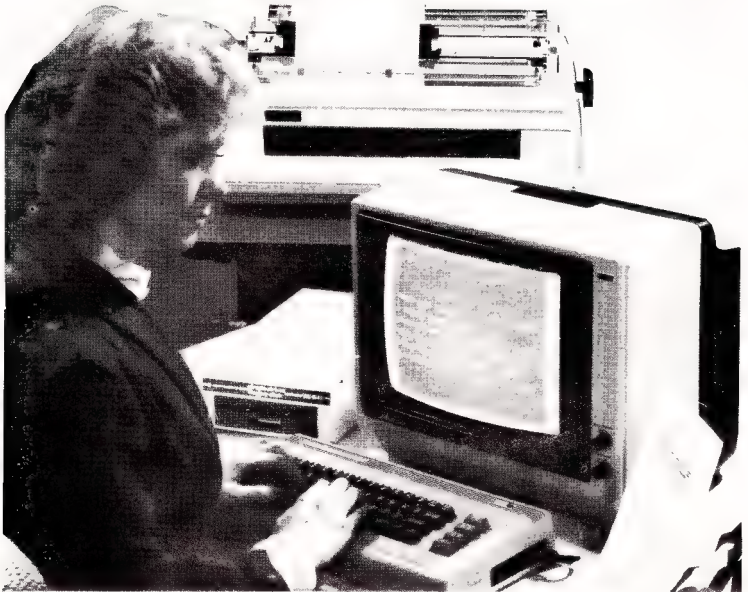
Son kayıttı, CPU, sayacın yardımıyla doğru yanıtları sayacak ve o boşluğa yazacak.

Birçok veri dosyaları, insanlar veya başka birtakım şeyler

hakkında toplanan bilgileri saklarlar. Bu dosyalar okullarda, hastanelerde ya da birçok insanı ilgilendiren işlerde vs. kullanılırlar.

Bunlar genelde çok büyük dosyalardır. Binlerce kaydı bir arada saklayabilirler. Sizin okulunuzda da şuna benzer bir dosya bulunabilir:

Adı Soyadı	Adresi	Yaşı	Sınıfı	Öğretmeni
Semra Akgül	Tuna Sok. 15/6	7	2	Ayşe Birgin
Ahmet Tanıl	Cumh. Cad. 6/4	9	4	Nesrin Çamoğlu
Sedef Şen	Gül Sokak. 5/8	10	5	Turgut Esin





Şimdi, diyelim ki Ahmet Tanıl'ın sınıfını öğrenmek istiyorsunuz. Bilgisayar, size şu mesajı gönderecektir:

ÖĞRENCİNİN ADI SOYADI:

Siz de öğrencinin adını ve soyadını yazacaksınız:

AHMET TANIL

Siz, öğrencinin adı ve soyadını yazdıktan sonra, bilgisayar size ikinci mesajını gönderecektir:

İSTENİLEN BİLGİNİN NUMARASINI YAZIN:

1. ADRESİ
2. YAŞI
3. SINIFI
4. ÖĞRETMENİ

Eğer siz de 3 diye yazarsanız, istediğiniz bilgiyi alırsınız:

AHMET TANIL SINIFI 4
Sonra, biraz önce verdiği

mesajın aynısını bir kez daha verecektir:

ÖĞRENCİNİN ADI SOYADI:

Siz, başka bir öğrenci hakkındaki bilgi almak istediğiniz sürece, bu iş sürekli yinelenir. Bilgisayar, istenilen bilgiyi verdikten sonra, otomatikman ilk verdiği mesaja döner. Devam edip etmemek sizin elinizdedir.

Size hangi bilgiyi istediğinizi soran ve onu veri listelerinden arayan şeyin; yazılan program olduğunu unutmayalım.

Giriş, mesaj geldiğinde bilgisayara gönderdiğimiz bilgilerdir. İşlem, istenilen bilginin bulunması ve çıkış da ekranda beliren bilginin kendisidir.

Birçok büyük mağazalar, stok listelerini buna benzer programlarla hazırlarlar.

Hastaneler, hastaları hakkındaki bilgileri aynı yöntemle saklarlar. Polis şubeleri suçluları

bu şekilde kodlar. Kütüphanelerde kitapların kaydı da bu şekilde tutulur. Bilgisayarın yardımıyla, alınan kitaplar, zamanında getirilmemiş kitaplar, atlanmadan kayda geçirilebilir.

Bu kayıtlar, aynı zamanda resim ya da grafik de saklayabilir. Bilgisayarın bu işi nasıl yaptığını anlayabilmek için, çıkış terminali ekranına bir göz atalım.

Terminal ekranı, binlerce küçük notadan oluşmuştur. Bu noktaların belirli bir düzene göre parlamasıyla, ekranda görüntü elde edilir. Yani bu küçük noktaların her birini birer lamba olarak düşünersek; biz bunların bir kısmını yakarak çeşitli harfler, sayılar, hatta şekiller elde edebiliriz.

Hangi karakter için hangi lambanın yanacağına biz karar veriyoruz. Ama bilgisayar, bu işi kendi yapıyor. Biz klavyeye istediğimiz karakteri bastığımız an, o





Video oyunlarının ilanları sık sık çeşitli dergilerde yayımlanıyor (üstte)...

karakter ekranda beliriyor.

Eğer ekranınızda sürekli olarak değişik karakterler kullanmak istiyorsanız -klavyede olmayanları kastediyoruz- onlar için özel tuşlar almanız gerekir.

Bilgisayar oyunlarındaki şekiller de, programlama yöntemiyle hazırlanır. Oradaki şekillerin sürekli hareket etmesini sağlamak için, programın içinde bir tur yaratılır. Bu turla, par-

layan noktaların adresleri sürekli değiştirilir.

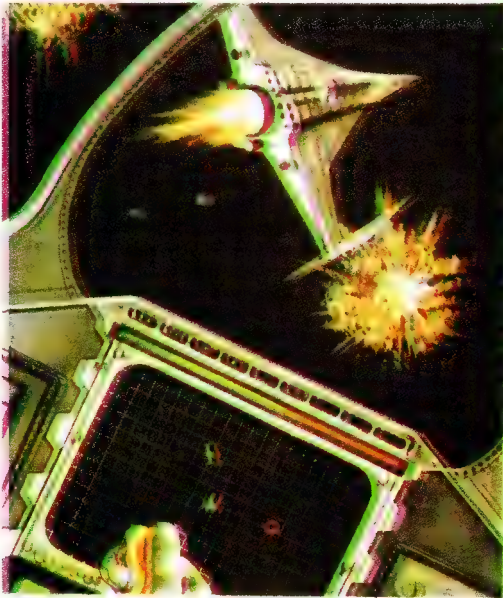
Eğer biz noktanın ilk adresine dokunmadan ikinci adresine sürekli 1 eklersek, nokta sağa kayıymış gibi görünür. Eğer tam tersini yaparsak, bu kez de nokta aşağıya kayıymış gibi görünür. Eğer iki adresi de değiştirirsek, nokta iki yönlü hareket eder. Ve bu adresler değiştikçe, bellekteki adresler de ona bağlı olarak değişir. Yani bilgisayar noktanın yerini her zaman bilir.

Bazı oyunlarda, şekilleri hare-

ket ettiren çubuklar vardır. Bunlar, kalemin yaptığı işi uzaktan yaparlar. Çubuğu sağa, sola, ileri ve geri hareket ettirerek, şekillerin de o yönlere doğru kaymasını sağlarız.

Siz de çubuğunuzu bu şekilde hareket ettirerek geminizi mayınların arasından geçirebilir, adamınızı labirentten çıkarabilirsiniz.

Çoğu oyunlarda, hareket eden düşman gemileri, mayınlar vardır. Bunların hareketlerini düzenleyen programlarda, bir-



Dergilerde çıkan oyun
itfani (ustte)...

çok değerlendirme komutu, turlar, SET ve RESET komutlarının yanında kullanılır. Düşmanınızın sizin hareketlerinize göre yer değiştirmesi bundandır. Değerlendirme komutları yardımıyla düşmanınız, sürekli sizi izler, mayınlarını geçebileceğinizi bölgelere aktarır.

Bilgisayarlarla oynanan oyunlar, her gün yeni bir şekle bürünüyor. İlk zamanlarda oynanılan "amiral battı"ların yerini sürekli yeni oyunlar alıyor. Artık çok çeşitli ve karmaşık oyunlar, yediden yetmiş kadar herkes tarafından oynanıyor. Ülkemizde yeni yeni yaygınlaşmaya başlayan bu oyunlar, gelişmiş ülkelerde yaşamın bir parçası haline gelmiş durumda.

Bilgisayar Kahramanları ve Canavarları

Kahraman ürkütücü, karanlık bir yerde uyanıyor. Kendini bir labirentin için de buluyor. Sağına baktığında, dar bir yol görüyor. Yolun üzerine beyaz lambalar sarkıyor. Lambalar oldukça parlak; ama labirent bir türlü aydınlanmıyor.

Kahraman, çevresine bakınıyor. Birçok duvar var. Buna karşın değişik yerlerden lambaların ışıkları görünüyor.

O da nesi? Hafif bir tıkırtı duyuluyor, sonra ses büyüyor, kahraman, panik içinde donup kalıyor. Sağına döndüğünde, canavarı görmeye ancak zaman

bulabiliyor. Canavar ahtapot, avını kovalamaya başlıyor.

Kahramanın kaçması gerekiyor. Ama nasıl? Bacakları yok ki.

Ancak hareket edebiliyor. Çevresindeki şeyleri yiyerek hareket edebiliyor. Ağzını açtı mı, yanındakileri bütünüyle yutabiliyor. Beyaz, parlak ışıkları, hatta ahtapotu bile yutabiliyor. Ancak ahtapotu yutabilmesi için, onu güçsüz bırakması gerekiyor.

Kahramanın canavar ahtaptan kaçıp kurtulabilmesi için, önünde sıralanan beyaz ışıkları yemesi gerekli. Sonra sıra büyük ışığa gelecek. Büyük ışığın boyu ise en az kahramanımız kadar.

Canavar gittikçe yaklaşıyor. Kahramanımız, olabildiğince çabuk bir şekilde lambaları yutmaya çalışıyor.

Canavar, hızla kahramanımıza doğru ilerliyor. Eğer onu yakalarsa, dişlerini geçirip onu balon gibi patlatacak. Gövdesi de bir balon gibi sönmeye başlayacak. Son bir titremeden sonra da yok olacak.

Kahraman, bu işi gerçekleştirmesine fırsat vermemek için, lambaları eskisinden de hızlı yutmaya başlıyor.

Büyük bir hızla önündeki lambaları yiyip, büyük lambaya erişmeye çalışıyor. Çünkü büyük lamba ona canavarı yiyebilecek gücü verecek. Ama yanlış yolu seçti. Canavar gittikçe yaklaşıyor. Son çırpınışlar bir yarar sağlamıyor. Kahramanımız yakalandı. Birkaç saniye içinde kayboluyor.

Canavar, üçe beş saniye sonra yeniden acıkıyor ve yeni bir av aramaya başlıyor.

Evet, yeni bir kahraman çıktı ortaya. Hem de tam eskisinin çıktığı yerden.

Ama bu kahraman daha akıllı. Ortaya çıkar çıkmaz, büyük bir hızla lambaları yemeye başlıyor.

Üç köşe dönüyor. Bu sırada arkasında üç tane ahtapot belirmiş. Hepsinin de amacı onu yok etmek. Ama bizimki akıllıca davranıp, onlara tuzak hazırlıyor. Birdenbire karanlık bir tünele

Dünyanın en çok satan mikro bilgisayarlarından biri olan "Sinclair" in, çocuklar için hazırladığı oyun programlarından birkaçı. (Sağda)

giriyor. Ahtapotlar da arkasından. Ama çok geç kalıyorlar. Çünkü o büyük lambaya ulaşılabile.. Şimdi artık onları yiyecek gücü var. Tünelin ağzından, onlara karşı pusuya yatıyor.

Bu kahramanlar ve canavarlar, 40 cm.'lik küçük, yassı bir labirente yaşıyorlar. Bu dünyanın tam üzerinde ise, parlak bir cam var. Bu cam iç dünyaya açılan bir pencere görevini almış.

Mavi duvarlı labirentin üzerinde büyük bir baş var. Bu baş büyük bir dikkatle içerisini izliyor. Labirentin mavi ve beyaz ışıkları, büyük beyaz dişlere yansıyor. Bu başın dudakları, müthiş bir gülümsemeyle kıvrılmış.

Labirentin altında, dev bir el, kırmızı kolu kavramış. Bu kol, beyaz bir kutunun üzerine yerleştirilmiş. Dev el, bu kolu sürekli bir aşağı, bir yukarı, sağa, sola hareket ettiriyor. Kol her oynayışında, içerideki kahraman da hareket ediyor. Yani, kahramanın hareketi bu kol tarafından düzenleniyor.

El, demin sözünü ettiğimiz başa ait. Bunlar bir insana ait. Kahramanı ahtaptottan kurtaran, büyük lambaya erişmeye çalışan hep o.

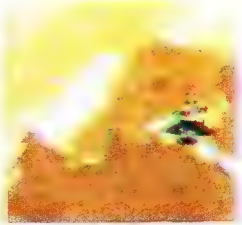
O, sizin yaşıncıda bir çocuk. Kahraman her lambayı yiyişinde, bir puan alıyor. Eğer ahtaptou da yerse, ona göre puanı daha da artıyor.

Yukarıda anlattığımız, bilgisayar oyunlarından yalnızca biri. Bunun gibi daha birçok oyun, büyük küçük herkesin elinde. Bazıları plastik küçük bir kutunun içine yerleştirilmiş. Bunları içeriye konulan bir elektronik beyin yönetiyor. Bu elektronik beyinler de "çip"lerden başka bir şey değil.

Bu oyunlar, evlerdeki televizyonlara bağlanabiliyor ve bir



Araba Yarışı



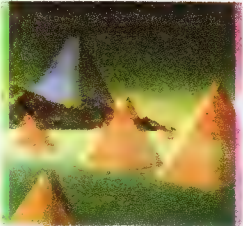
Uzay Savaşı



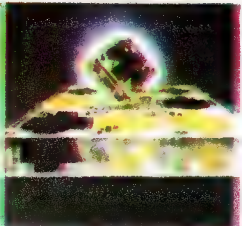
Bil-Yap



Uzayda Yaşam



Piramit Oyunu



Küp Oyunu

düğüme basmakla ekrana canavarlar, uçak gemileri, bombalar ve daha başka birçok şey çıkarılabilir.

Evinizde, yalnız başına kaldığınızda canınız sıkılmadan bu oyunlara dalabiliyorsunuz. Ya bir mikrobilgisayarın başında, ya da televizyonunuzun önünde, ya da tek parça olarak satılan bilgisayar oyunlarıyla istediğiniz dünyaya girebiliyorsunuz. Föbol maçları, uzay oyunları elinizin altında oluyor.

Şimdi size bir bilgisayar oyununun nasıl tamamlandığı hakkında fikir verecek olan bir öykü anlatacağız. Bu öykü gerçektir.

Yıldız Kalesinin Doğuşu

Wynn Bailey, resim çekebilecek bilgisayar kameraları üzerine çalışıyordu. 1970'lerin ortasında, bilgisayar oyunları ilgisini çekti. Kendi oyununu kendisi yazmak isteyince, masasının başına geçti. Kısa bir süre sonra işinden ayrılıp, bilgisayar oyunları üzerinde çalışan bir şirkete geçti.

Scott Boden, genç, çekingen bir bilgisayar programcısıydı. Bailey ile birlikte yeni oyunlar geliştirmek üzere çalışıyorlardı.

Boden, Bailey'in fikirlerini program haline getirmekte ustaydı.

Birgün, Bailey Boden'e yepeni bir oyundan sözettii. Boden'i iŖe baŖlatmak iin, onu byk bir karatahtası olan bir odaya gtrd. Bailey eline tebeŖiri aldı ve uzayın derinliklerinde dolaŖan bir yildız kalesi izdi. Kalenin ortasında byk bir top vardı. Bailey, kalenin evresine daireler izdi. Bu daireler, kaleyi koruyan enerji halkalarıydı. Bu halkaları yok etmeden kaleye Ŗaldırmak olanaksızdı.

Bailey, kalenin epeyce uzağına kk bir gen izdi. Bu oyuncunun gemisi olacaktı. Oyuncunun yapması gereken enerji halkalarını yok edip, arkasından kalenin iindeki topu patlatıp, kaleyi havaya rmak-tı.

Boden, Bailey'in izimi zerinde alıŖtı. Oyun, ok kolay bir Ŗeye benziyordu. Oyuncu, iki tuŖ (SAĞ ve SOL) kullanabiliyordu. Bu tuŖlarla enerji halkalarını hedefleyip vurmaŖı gerekiyordu. Bir de ateŖ etmesini saėlayan "ateŖ" tuŖu vardı. ok kolay bir oyun gibi grnyordu. İyi bir Ŗekilde ayarlanabilirse. Sonra Bailey, yznde muzip bir glmsemeyle oyunun gerisini anlattı. Boden yavaŖ yavaŖ Bai-

ley'in oyun hakkındaki gerek planını kavradı. Ve oyunun hide o kadar kolay olmadıėını grd. ok zor bir oyundu bu. Zor olduėu kadar da heyecanlı...

Boden, bilgisayarının baŖına geti. Komutları yazmaya baŖladı. Bailey'in dŖncelerini bilgisayarın anlayacaėı bir Ŗekle evirdi. Bazı komutlar kalenin, enerji halkalarını ve topu izmek iin kullanılmıŖtı. Diėerleri oyuncunun gemisini ve vereceėi komutlara karŖı nasıl hareket edileceėini gsteriyordu. Geri kalanlar da, patlamaları ve oyun sırasında ıkacak diėer sesleri ayarlıyordu. Ama bu komutların oėu, Bailey'in dŖncelerinin bilgisayar diline aktarımıydı. Bilgisayarın oyuncuyu yenmek iin kullanacaėı kendi silahları.

Bu silahlar nasıldı? Niin bu oyun o kadar zorduy?

Birincisi, Yıldız Kalesi'nin topu yle sessiz durmuyordu. Srekli dnyordu. Bu da geminin srekli hareket etmesine yol aıyordu. Eėer gemi hareket etmezse, top onu anında yok ederdi. Bunun iin enerji halkalarının arasında ufak bir aık yeterliydi. Top, hibir zaman boŖa atıŖmıyordu; ve atıŖları ldrcyd. Yani bir kez yara

alan geminin kurtulma Ŗansı kalmıyordu.

Bu arada akıllı mayınlar da vardı. Bailey bunları Boden'e gsterirken, dıŖ halkaların zerine X'ler izmiŖti. Bu X'lerden her birinin nkleer bir bomba kadar tahrip gc var. Halkalar dndkce bu mayınları oluŖturup uzaya fırlatıyorlar

Peki bu mayınlar neden "akıllı"? Bunlar, halkadan ayrılır ayrılmaz, sizin geminizi aramaya ıkıyorlar. abucuk da sizi bulup zerinize doėru hızla gelmeye baŖlıyorlar. Eėer size arparsa, vay halinize. Anında havaya yosunuz.

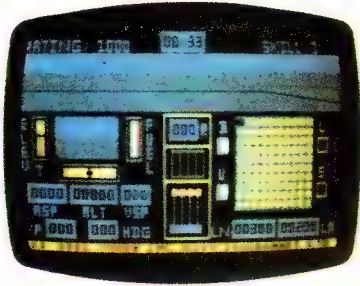
Bundan kaıŖ yok. Ekranın zerinde istediėiniz yere gidebilirsiniz. Ayrıca bir dğmeye basıp mthiŖ bir hız da kazanabilirsiniz. Ama bu akıllı mayınlar hibir zaman pes etmezler. Srekli olarak sizi kovalamaya devam edeceklerdir. Kesinlikle izinizi kaybetmezler. Srekli yaklaŖırlar size.

Ne yapabilirsiniz?

Neyse ki,  tane geminiz var. Eėer akıllı mayınlardan biri ilk geminizi yok ederse zlmeyin. İki tane daha geminiz var.

Bundan daha iyisi, bu mayınlar enerji halkalarından ıkma-dan onları ateŖlerseniz, onlar Ŗizi





"Atari" firmasının çıkardığı en son oyunlardan biri: Jumbo Jet Pilotu (sağda) .. Bu oyunda oyuncu televizyon ekranında beliren bir uçağın kontrol tablolarının yardımıyla, uçağı havada tut-

mayı deniyor. Bu arada uçak çeşitli hava koşullarıyla da karşılaşiyor. Soldaki oyunda ise, oyuncak tropik bir ormanın içinde kayboluyor.

yok etmeden siz onları yok etmiş olursunuz. Ayrıca, eğer iyi nişancıysanız, onları üzerinize doğru gelirken de ateşleyebilirsiniz.

Bu mayınlar ne kadar akıllı olurlarsa olsunlar, yalnızca Boden'in onlara gösterdiklerini yapabilirler.

Mayınlar, sizin yerinizi biliyor gibi görünür. Ama bu çok karmaşık ve pahalı bir iştir. Sizin geminiz, saniyenin otuz sekizde biri gibi bir zamanda ekranın bir başka tarafına geçebilir. Bir düzine mayının sizin izinizi sürmesi, çok daha fazla devreyi, belleği ve beyin gücünü gerektirir. Her saniyenin otuz sekizde biri bir zaman içinde, küçücük bir beynin her mayına sizin yeni yerinizi söylemesi gerekir. Bu da yapılan oyunu çok karmaşık bir hale getirdiği gibi tutarını da yükseltir.

Sonuç olarak mayınlar, sizin nerede olduğunuzu bilmez. Bunun yerine tahmin etmeye çalışır. Onların dilinde yani. Çünkü biliyorsunuz, bilgisayarlar kendi kendilerine düşünemezler. Biraz önce de söylediğimiz gibi, programcının yazdıklarının uygularlar.

Her mayın, kendine göre bir



Gorf



Yüzbaşı Cosmo

yol izler. Bu yolları, hapishanelerin çevresine yerleştirilen projektörlerin yollarına benzetebiliriz. Eğer dikkatle onları izleyebilirsiniz, izledikleri yolu çıkartabilirsiniz. Sonra da hapishaneden kaçmaya çalışan bir mahkûm gibi, bundan yararlanabilirsiniz. Mayınların bir yöne doğru kıvrılmasını bekler, sonra da tam ters yöne doğru hareket edebilirsiniz. Bu arada enerji halkalarını da yok eder ve topa hedef olmandan kendinizi korursunuz

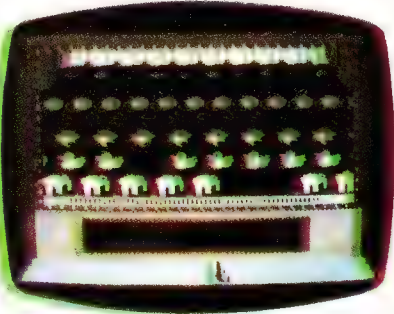
Nasıl, oyunu beğendiniz mi? Bu ve benzeri oyunlar, şu anda birçok yerde kâpışılıyor. Bunun gibi zor olanları ise daha zevkle oynanıyor. Çünkü bir iş ne kadar zor olursa, onun üzerinde ne kadar çok çalışılırsa, sonunda elde edilen zafer de o denli tatlı oluyor.

Bilgisayar oyunlarının eğlendirici olmasının yanında öğretici yanı da var. Şu anda bazı işletmeciler, bilgisayarlı oyuncaklar da yapıyorlar. Bu oyuncakların bazıları matematik problemlerini içeriyor. Bazıları da yazım kurallarını iyi bilmeyi gerektiriyor. Oyuncu rastgele bir soru seçiyor. Soruyu yanıtladıktan sonra bilgisayar, ona doğru ya da yanlış yanıt verdiğini söylüyor.

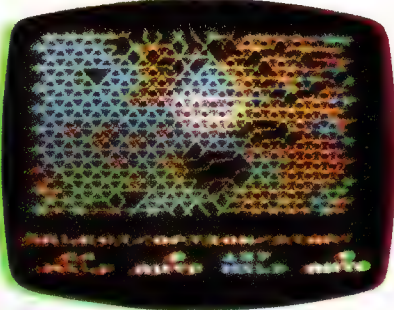
Bunların daha da gelişmişleri, sözlük görevini görüyor. Özellikle dil bilmeyen turistler için çok yararlı oluyor. Turist, öğrenmek istediği kelimeyi kendi dilinden yazdığı anda, ona istediği dilden karşılığı yazılıyor.

Bazı bilgisayar oyunları ise, birtakım tarihsel olaylarla ilgiliydi. Bilgisayar, oyunculara bu olaylar hakkında bilgi veriyor ve aynı zamanda orada olsalardı yapabilecekleri şeyleri açıklıyor. Bu sayede oyuncu hem eğlenirken, hem de bu olaylar hakkında bir şeyler öğreniyor.

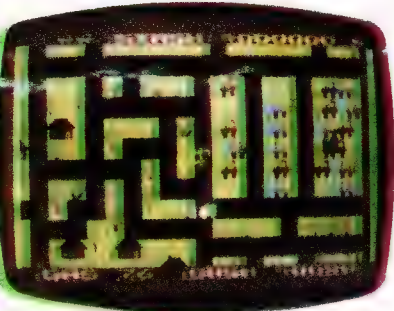
En yaygın video oyunlarından üçü. (Sağda)



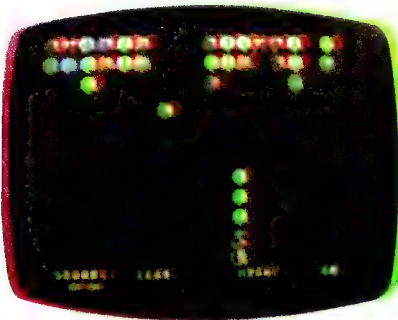
Atış Poligonu



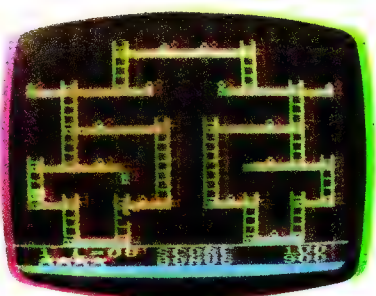
Böcekler



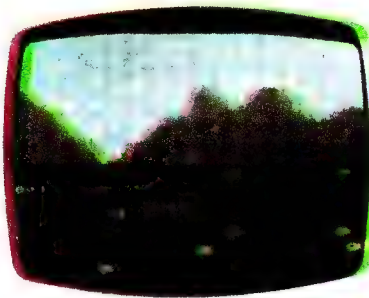
Kaçış



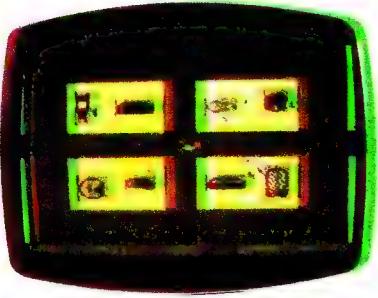
İğne Deliği



Atlayıcı



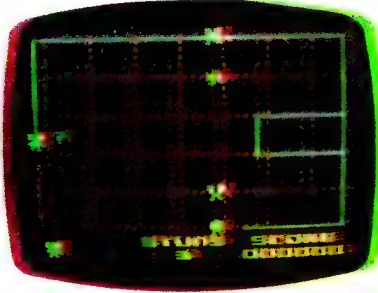
Yıldız Mezarlığı



Piknik Oyunu



Futbol



Saklambaç

Bilgisayar Oyunlarını Kendi Yararımıza Oynayalım

BUNALDIĞIMI hissettiğim zaman, hemen bilgisayarımın başına koşuyorum. Birkaç oyun oynamak, kanımı yeniden harekete geçiriyor.

Bunları söyleyen, genç bir yönetici. Evet, bilgisayar oyunları yalnızca çocukların oynamadığına bir kanıt. Bu genç adam, işini de ciddiye alıyor.

Ama oyunların zaman öldürmek uğruna yapılan birçok şeyden çok daha anlamlı olduğunu farkeden çok kişiden biri.

Bilgisayar oyunlarında, video oyunlarından çok daha fazla bellek kapasitesi olduğu için, yapımcılar çok karmaşık oyunlara giriştiler. Sonuçta, şimdi birçok bilgisayar kullanıcısı saatlerini bu aletin başında geçirip, yaratıklarla savaşıp "dünyayı kurtarmaya" çalışıyorlar.

Bu oyunlarla oynayanlar vakitlerini mi öldürüyorlar? Uzmanlar bu görüntüde değil. Eğitim uzmanları ve psikologlar, bu oyunların mantıklı düşünme yeteneğini arttırdığını ve problem çözme yeteneğini geliştirdiğini söylüyorlar. Yani bu oyunlar, kişiye gerçek yaşamında da başarı kapılarını arıyor. Ayrıca akıllıca hazırlanmış oyunlar, öğrencilerin dikkatlerini sürekli canlı tutup, öğrenmelerini çabuklaştırıyorlar.

Bir soru daha var: İnsanlar niçin bu oyunları oynuyorlar? Kendilerini aslında var olmayan bir dünyayla avutmak için mi? Yoksa yarışmayı sevdikleri ve yaşamdan bunaldıkları için mi? Her insan, değişik bir amaçla oynuyor bunları.

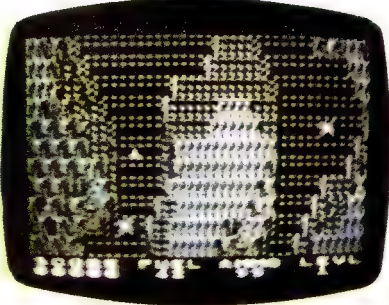
Dağların arasında, ahşap bir evdesiniz. Evin elektriği yok. Dışarıda yaban domuzları, geyik ve yılanlar var. Yakında, alabalıklarla dolu bir dere var. Şimdi çevrede kimse yok, ama yakında kaçak avcılar kovalamaya başlamanız gerekiyor. Elinizde bir tabanca var.

Bu daha çok, maceralı bir oyunun tanıtımına benziyor, değil mi? Ama gerçekte Amerika Birleşik Devletleri üniversitelerinden birinde profesörlük yapan bir kişinin evi. Bu profesör, insanların niçin bilgisayar oyunları bu kadar çok ilgilendiklerini araştırıyor.

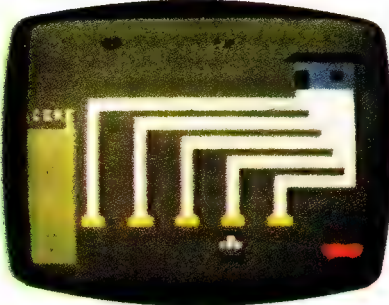
İnsanların bilgisayar oyunları bu kadar yakından ilgilanmesinin bir nedeninin, bilgisayarın insanların bütün dikkatini alması, onları sürekli tetikte tutması; ama aynı zamanda da rahatlatmasıdır.



Deniz Canavarı



Trion



Kümes Oyunu

Aynı profesör, bir oyunda ne kadar yarışma hissi varsa, o kadar çok tutuluyor. Çünkü günümüzde, bütün yaşam bu tip bir yarışma üzerine kurulmuş. İnsanlar oynarken, tutukları takımın heyeenah bir maçındaymış gibi davranabiliyorlar. Yani epey çekışmeli bir hale getiriyorlar oyunu. Hatta hiç yarışma eğilimi olmadığını iddia eden insanlar bile, bilgisayarın karşısına geçince onu yenmek için ne gerekirse yapıyorlar.

Bazı insanlar, bu oyunları bambaşka nedenlerle oynuyorlar. "Yorgunluk atmak", "rahatlamak" için değil, "bambaşka bir dünyaya geçmek için" oynuyorlar. Bazı insanların en sevdiği müzikten söz ettikleri bir şekilde söz ediyorlar bilgisayardan. Nasıl ki insan sevdiği bir müziği dinlemekten mutluluk duyar, bu insanlar da aynı şekilde bilgisayar oyunları oynamaktan mutluluk duyuyorlar.

Üretilen her yeni üründe yer alacak maddelerin bileşim oranları bilgisayar tarafından saptanarak, verimliliğin artırılmasına çalışılır. (Altta)



Akıllı Makinelerle Dolu Bir Dünya

Bütün bilgisayarlar kol
saati ve oyunda mı
kullanılıyor?

Kesinlikle.

Minicik bilgisayarlar, günlük işler için kullanılan binlerce aletin içinde. Bunlar, makineleri daha ucuz ve güvenilir yapmak için kullanılan beyinler durumunda. Radyolarda, hesap makinelerinde, televizyonlar, telefonlarda, mikrodalgalı fırınlarda, bulaşık, çamaşır makinelerinde ve arabalarda, her yerde bilgisayar çipleri kullanılıyor.

Uzmanların, gelecekteki dünya için daha ilginç görüşleri var. Yakında gözlüklerin, kredi kartlarının, mobilyaların ve giysilerin içinde de boy gösterecek bu çipler. Şimdiden bazı sakat insanların vücuduna girmiş durumdadılar.

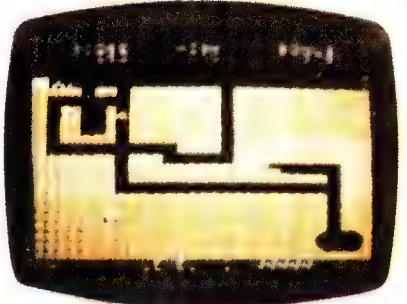
Yeni araştırmalar sonucu üretilen bazı çipler, gözleri görmeyen insanların görmesini sağlayacak. Bu çiplerle yapılan deneyler, şu anda laboratuvarlarda sürmekte. Bu çipler, ışığı emebiliyor. Sonra da onu doğrudan beyne gönderiyor. Beyin de, normal bir gözün algılayabileceği şekle benzer bir şekli görmüş oluyor. Bu belki tam olarak bir "görme" değil, ama görmeyen insanlar için bir umut ışığı olsa gerek.

Dünyamız, yepyeni bir hızla ilerliyor. Rock müzik, artık bilgisayarlaşmış aletlerle çalınıyor. Ve yine bilgisayar yardımıyla kayda geçiriliyor. Sinemada bazı sahneler, bilgisayar yardımıyla çekiliyor. Bazı aletleri, olimpiyatlara hazırlanırken bilgisayarlar çalıştırıyor. Bu bilgisayarlar, atletin ayakkabısının altına yerleştiriliyor. Atlet, çalışma sırasında, onun hakkındaki tüm bilgileri, bu bilgisayar çıkarıyor, not ediyor.

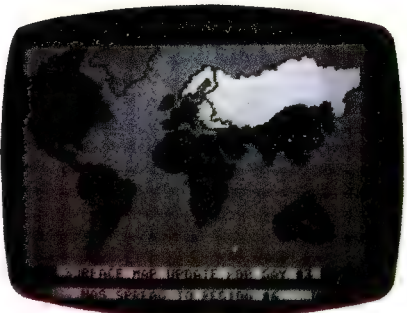
Kulakları duymayan insanlar da, bilgisayar yardımıyla "tele-



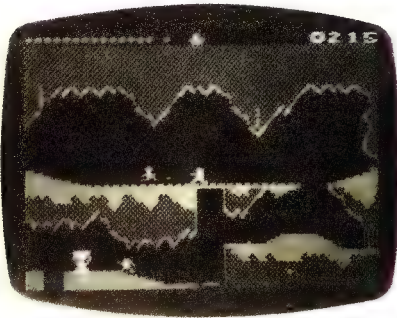
Kıyamet Kalesi



Firavunun Mezarı



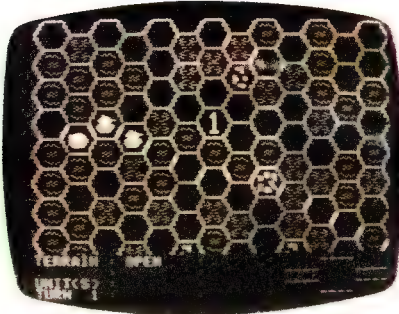
Tiyatro



Tünel Oyunu



Uzay Mikrobu



1985 Almanya

fonlaşabiliyorlar". Bu iş için kullanılan makine daktiloya benziyor. Bilgisayar haberleşme yöntemiyle birbirleriyle iletişim kurabiliyorlar.

Bilgisayar, daha pek çok kişiyi mutlu ediyor. Yatağından kalkamayacak kadar hasta olanlar ya da işyerine gelemeyecek durumda olan kimseler de, yine bilgisayar yardımıyla çalışabiliyor. Hepsinin birer bilgisayarları var. İşyerleriyle bu şekilde haberleşebiliyorlar. Bilgisayar, onlar için elektronik güvercin görevini yapıyor. Haberleşme, telefon telleri yardımıyla yapılıyor. Genelde bu tip çalışanları olan şirketler, bütün bir ülkeye, hatta dünyaya yayılmış oluyor.

Daha başka yerlerde bilgisayarlar doktorlara yardımcı oluyorlar. Robotların içine yerleştirilen bilgisayarlar, fabrikada işleri kolaylaştırıyor. Bu tip yerlerde çalışan sayısını azaltan bilgisayarlar, apayrı bir iş kolunu oluşturuyorlar. Çok değil, beş on sene önce bilgisayar programcısı kelimesi pek çok kimse için bir şey ifade etmiyordu. Oysa şimdi birçok şirket bünyesinde bilgisayar programcısı bulunduruyor.

Başka yerlerde bilgisayarlar yerbilimcilere ve denizbilimcilere yardım ediyor. Petrol araştırmalarına katkıda bulunuyor. Bilgisayar kontrolünde robotlar yerin yüzlerce metre altında maden yataklarını arıyor. Yine aynı işi gören robotlar, okyanusların buz gibi suyunda çalışıyor.

Artık küçük bilgisayar çipleri her yerde, denizin altında, başımızın üstünde, giysilerimizin üzerinde ve hatta vücudumuzda. Bazı insanlar, bilgisayarları hâlâ hantal, büyük makineler olarak düşünürler. Artık o model bilgisayarlar kalmadı, ama yaptıkları işler gittikçe büyüyor.

Bugün, yurdumuzda bilgisayar yeni yeni kullanılmaya başlanmış durumda. Ama hesap makinelerini her yerde görmek olası. Hesap makineleri de bir çeşit bilgisayardır. Dilerseniz, biraz da hesap makinelerine göz atalım.

Hesap Makineleri

HESAP makineleri, matematiksel işlemlerin yapılmasında en çok kullanılan aletlerden biridir. Bazıları, bir kol saatine sığabilecek kadar küçüktür. Ancak bunların da şöyle bir zorluğu var: Tuşları o kadar küçük ki, normal bir parmak onları kullanmakta zorluk çeker!

Temel Parçalar: Hesap makinelerinin üç temel parçası vardır. Bunlar, güç kaynağı, klavye ve ekrandır. Çoğu hesap makinesi pille çalışır. Ama prize takılabilecek ek kordonları da bulunur.

Genellikle hesap makinelerinin ekranı sekiz basamaklı bir sayıyı alabilecek büyüklükte yapılır. Daha gelişmiş olanlar ise, on ya da on iki basamak gösterebilir. Ayrıca sayının artı ya da eksi olduğunu göstermek için bir yer, ve "kayan" ondalık virgüli vardır. Bu virgülin kayması, ekranın herhangi bir yerinde gösterilebilmesi anlamına gelir. Yani hesap makinelerinde, örneğin bir 0,78 sayısı; 0,078x10 olarak da gösterilebilir.

Her basamak, yedi küçük doğru ile gösterilir. Eğer bütün doğrular yanarsa, 8 rakamı okunur. Eğer ortadaki doğru yanmazsa, 0 okunur. Bu gibi değişimler yaparak bütün sayıları yazmak olasıdır. Ekran da iki çeşittir. Bir çeşidinde, sıvı kristal kullanılır. Diğeri ise daha eski bir yöntem olan ışık verici diyet kullanmaktır. Diyetler, koyu bir zemin üzerinde kırmızı ya da yeşil ışık verirler. Buna karşın sıvı kristaller, açık bir zemin üzerinde sayıları siyah gösterirler.

Hesap makinesinin klavyesi üzerinde yer alan tuşların sayısı, makineden makineye değişir. Ama hiçbir makineninki 40'ın üzerinde değildir. Klavyenin orta yerinde dokuz rakam bulunur. 1,2,3,4,5,6,7,8,9, aşağıdan yukarıya doğru belli bir düzen içinde yazılır.

Diğer tuşlara fonksiyon tuşu

Hesap makinesi alırken, gerek duyduğunuz fonksiyonların bulunmasına dikkat ediniz. Ama gereksiz büyüklükte bir makine alıp savurganlık da yapmayınız. (Sağda)

Kullanıldıktan sonra makineyi kapamak çok önemlidir. Böylelikle hem daha az enerji kullanılır, hem de makinenin ömrü uzar. Bazı makineler, otomatik olarak belirli bir süre kullanılmayınca kapanır.

Özel fonksiyon tuşları, fazla kullanılan işlemleri kısa kesmek için konulur. Özel makinelerin, daha çok fonksiyon tuşları vardır. Bunlar kullanıcıya büyük kolaylık sağlar.

Bu tuş, bellek dışında tüm bilgileri silmek için kullanılır. Eğer uzun bir işlem sırasında yanlış yapırsanız, bir üstteki tuş yalnızca son yazılan sayıyı siler. Geride kalan işleme dokunmaz.

Ekran, genelde 8 ya da daha çok rakam alır. Işık veren diyetler hem ucuzdur, hem de karanlıkta görülebilir. Ancak sıvı kristaller daha az enerji harcar.

Sayılar hep aynı düzende yazılır. 0'ın ve kesir virgülünün yerleri değişik olmasında birlikte, onlar da hep sayıların altındadırlar.

adı verilir. Genelde, çoğu hesap makinelerinin işlemin sonucunu göstermeye yarayan bir tuşları vardır, "=" işaretiyle gösterilir.

Ancak bu tuşun bütün hesap makinelerinde bulunması gerekmez. Bazı makineler, işlem bittikten sonra başka bir tuşa basmaya



Hesap makinelerinde, genellikle pil kullanılır. Ama bazılarında ek olarak şehir ceryanını kullanmak için özel bir fiş vardır. Bazılarında ise, yine şehir ceryanıyla doldurulabilen piller vardır.

Kimizi tuşları, bellek ile ilgili işlemler için kullanılır. Belleğe bir sayı toplamak ya da çıkartmak isteniyorsa, M + ya da M - tuşları kullanılır. Belleği çağırarak için MR tuşuna, silmek için de CM tuşuna basılır. Birçok hesap makinesinin belleği, birkaç sayıyı birden saklayabilecek kapasitededir.

hesap makinelerinin belleği, makineyi kapatır kapatmaz silinir. Ancak yeni çıkmış olan gelişmiş modellerin böyle bir sorunu yoktur.

Günlük kullanımda yer alan basit, ucuz hesap makinelerinin yanında, özel birtakım işler için hazırlanmış bilgisayarlar da vardır. Bilim adamları, işadamları ve mühendisler için özel olarak hazırlanmış hesap makineleri de vardır.

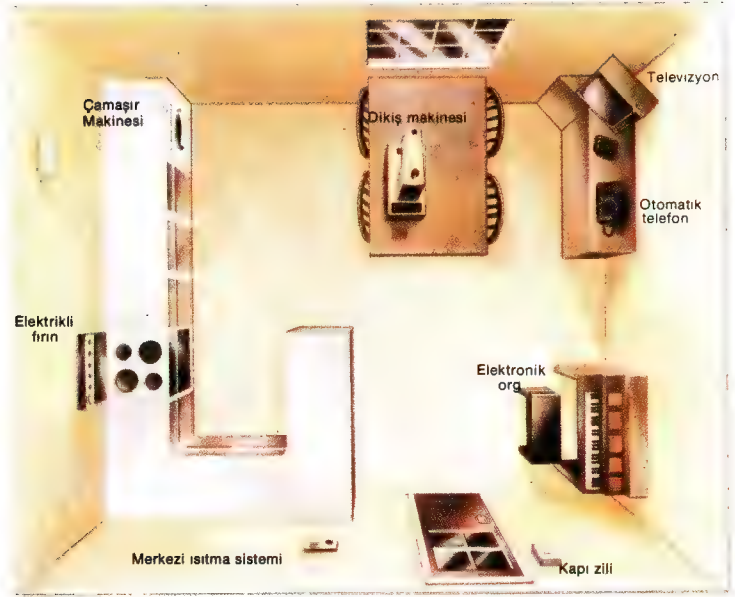
Bazı makineler ise, küçük programlar alabilecek kapasitededir. Bunlar bir mikrobilgisayar görevini üstlenebilirler. Yapımcılar, biyotiritimden gökbilime, denizaltı çalışmalarına kadar çok çeşitli konularda paket programlar hazırlamışlar ve piyasaya sürmüşlerdir. Hatta, yüzme havuzlarına ne kadar klor konulacağını belirleyen özel bir hesap makinesi de vardır.

Basit aritmetik fonksiyon tuşları. Eğer bağlantıları iyi yapılmamışsa, bu tuşlar çabuk bozulabilir ve yanlış işlemler yapabilir. Bazı makinelerin tuşları çok hassas yapılmıştır. İkinci kez basıldığında uyarı sinyali verir.

gerek olmadan sonucu yazarlar.

Bellek: Çoğu hesap makinesinde, bellek de vardır. Bu işlemlerde fazla sayıda kullanılacak

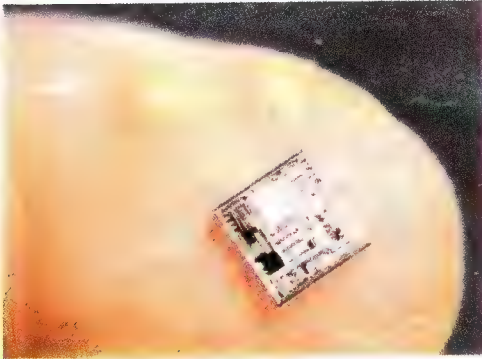
olan sabitleri yazmak için ya da daha sonra gerekecek olan bir işlem sonucunu saklamak için kullanılır. Fazla gelişmiş olmayan



Mikroişlemci ile çalışan bazı ev aletleri. Bunların bu kadar yaygınlaşmasının bir nedeni de

toplu olarak üretildikleri zaman çok ucuza çıkmalarıdır. (Üstte)

Bir santimetrekareden küçük bir çipin içine yüzlerce devre elemanı sığabilir. (Solda)



Mikroişlemciler

Programlanabilir her aletin içinde bir mikroişlemci vardır. Bu isimden şimdiye kadar epeyce söz ettik. Ama doğru dürüst açıklamasını yapmadık. Mikroişlemciler, küçük olmalarına karşın pek çok görevi üstlenebilirler. Bunların en büyük kolaylığı da, bir anda çok sayıda üretilebilmesidir.

Çipler: Mikroişlemcinin temeli, bir çiptir. Bir santimetrekareden daha küçük olabilir. Değişik amaçlar için kullanılan çipler vardır. Bir de genel kullanım için hazırlanmış çipler

vardır. Bunlar, çeşitli işlerde kullanılabilirler. Tek bir amaç için ve az sayıda üretilen çipler, çok pahalıya malolabilir.

Güvenilirlik: Mikroişlemcilerin içinde hareket eden hiçbir parça yoktur. Bu yüzden aynı işi yapan mekanik aletlerden daha güvenilirlerdir. Ayrıca çok daha küçüktürler. Mikroişlemciler, bir zamanlar tamamiyle mekanik olan bazı aletlerin içine bile yerleştirilmeye başlanmıştır. Kasalarda, benzin istasyonlarında artık hep mikroişlemciler kullanılmaktadır.

Mikroişlemciler, evin içine de girmeyi başardılar. Çamaşır makineleri, fırınlar, dikiş makineleri, fotoğraf makineleri ve hatta kapı zilleri bir mikroişlemciyle kontrol edilir duruma gelmiştir. Fırının ve çamaşır makinesinin içindekiler, otomatik çalışma koşullarını arttırıyor, kullanıcıya daha geniş seçenekler tanıyor. Dikiş makinelerindekiler ise, çok değişik nakışları, hatta düğme dikmeyi makineye yaptırıyor. Bazı işletmeciler ise, bu aletlere ek işler yaptırabilecek özel mikroişlemcileri piyasaya çıkartmış durumda.

Endüstride Mikroişlemciler: Araba endüstrisi, mikroişlemciyi kullanan ilk endüstri kolu oldu. Şimdi birçok yapımcı, fabrikalarında bazı işleri özel robotlara yaptırıyor. Ayrıca mikrolar, arabanın içine de el attı. Benzin kullanımında tasarrufa gitmek amacıyla, bujiye yerleştirdiler.

Şimdi, endüstrinin diğer kollarında da mikroişlemciler oldukça yaygınlaşmış durumdadır. Mekanik ve elektrikli aletlerin yerini her geçen gün yeni bir mikro alıyor. Tekstil endüstrisinde, dokuma makinelerine takılan mikroişlemciler ise, kumaşın dokunuşunu kontrol ediyor.

“El değmeden” hazırlanıp satışa sunulan fabrika ürünleri de, mikrolar tarafından kontrol ediliyor.



Londra Hayvanat Bahçesinin dev pandası Ching-Ching-Ching, 1980 yılında bir ameliyat geçirmişti. Veteriner, iyileşene kadar onun bacağına mikroşlemci kontrolünde bir alet taktı. (Üstte)

Mikroşlemci üreten bir teknik eleman. Her şeyin çok temiz olması gerekiyor. Çünkü istenmeyen toz parçaları ve kimyasal maddeler, çiplerin çalışmamasına neden oluyor. (Altta)



Okulda



Mikrobilgisayarlar: Artık her okul kendine bir bilgisayar alma yolunda. Ayrıca bilgisayar elemanı yetiştiren okullar da gün geçtikçe artıyor. Çoğu işyeri, bilgisayar kullanmaya başladığı için, buralarda çalışacakların bilgisayarı öğrenmesi gerekiyor. Elektronik ve programlama, en önemli derslerden sayılmaya başlandı. Okullarda bilgisayar,

diğer konuları öğrenmekte de öğrencilere yardımcı oluyor. Matematik öğrencileri bilgisayarla problem çözerken, yer bilim öğrencileri harita yapmasını öğrenebiliyorlar. Programlama tekniği arttıkça, bu konular da çoğalacak. Ayrıca okullar bilgisayarları kendi yönetim işlerinde de kullanabiliyorlar. (Üstte)

Mikrobilgisayarlar

BİR süre önce sözünü ettiğimiz bilgisayar oyunları, mikrobilgisayarlarla oynanır. Mikrobilgisayarların kapasitesi, normal bilgisayarlara göre düşüktür, ama az yer kaplar ve kullanımı çok daha kolaydır. Evlerde kullanılan bilgisayarlarda mikrobilgisayarlardır.

1950'lerin, 1960'ların bilgisayarlarnı çalıştırmak için çok fazla insana gereksinim duyuluyordu. Ayrıca bu makineler çok pahalıydı ve özel havalandırılmalı odalarda kullanılması gerekiyordu. Şimdilerde ise, bir kişinin birkaç aylık maaşıyla bir bilgisayar alınabiliyor. Bu mikrobilgisayarlar, normal koşullarda çalışan bir işyerinin çalışmalarını düzenleyebilecek kapasitede yapıyor.

Dolayısıyla, evlerde de kullanılabiliyor.

Bu, mikrobilgisayarları bir anda gelişen bir kol durumuna getirdi. Bilgisayar endüstrisinde en çok üretim yapan kuruluşlar, mikrobilgisayar kuruluşlarıdır. 1970'lerde, birçok işletmeci, yapımçı, kendi bilgisayarını üretmeye başladı. Aynı zamanda diskler, yazıcılar ve diğer yan üniteler de satışa sunuldu.

Mikro ve Mini: Mikrobilgisayarlar, bir ya da birkaç mikroişlemci çipiyle birkaç RAM çipinden meydana gelir. En basit modellerin ise, hiçbir yan üniteye gereksinimi yoktur. Bu tip mikrobilgisayarlar, normal kaset teypte, kasedini bilgi girişi ve depolama yapmak için kullanabilirler. Bilgi çıkışları ise televizyon ekranından verilir. Bu gibi maki-

Çiftlikte



Mikrobilgisayarlar için çiftlikte yapılacak pek çok şey vardır. Bazı çiftlikler, her inneğin kaç kilo süt verdiğini her gün kayda geçirirler. Bu işi bilgisayar yapar ve çiftçiye hangi yemeğin üretimi arttırdığını bulma konusunda yardımcı olur. Ayrıca, çiftlikle ilgili diğer gereken hesaplamaları da yapar. (Sağda)

neler, bilgisayarlara ılgısı olan, onlarla amatörce ilgilenen kişilere satılır.

Büyük bir bilgisayar alacak kadar gelişmiş olmayan, küçük işletmeler de, işlerini yürütmek için bir mikrobilgisayar alabilirler. Bunun için özel olarak hazırlanmış bilgisayarlar vardır. Muhasebe işleri için ya da mühendislik işleri için hazırlanan mikrobilgisayarlar vardır. Hatta bunların bazıları o kadar geliş-

miştir ki, onlara artık mikro denmez. Mikrobilgisayarlardan biraz daha kapasiteli bilgisayarlara minibilgisayar adı verilir.

Program Paketleri: Küçük işletmeler, bilgisayar kullanmak istiyorlardı. Bilgisayar hakkında pek bir şey bilmiyorlardı. Bunun üzerine bilgisayar şirketleri kendi

bilgisayarları için programlar hazırlamaya başladı. Bu programlar, genel olarak istenen her konuyu kapsıyordu. Örneğin muhasebe programları, işletmecilerin en çok istediği programlardan biridir. Bu şekilde, kullanıcının herhangi bir dili bilmesine gerek kalmıyor. Kendine program hazır olarak geliyor.

Dış Kliniğinde



Dış doktorlarının işlerini kolaylaştırmak için özel bazı programlar hazırlanmıştır. Her hastanın dişlerinin durumu ve ne bakımları gördüğünü

saklayabilir. Randevular, ve toplam ücreti hesaplamak ve hastaya bir sonraki randevu tarihini göndermek için kullanılabilir. (Üstte)



Ona da yalnızca gereken bilgilerin girişini yapmak kalıyor. Bu da kullanıcıya büyük kolaylık sağlıyor.

Bu programlara toplu halde program paketleri adı verilir. Çünkü bunlar programcıya bir paket halinde verilir. Kullanıcı, istediği programı alır, hiç uğraşmadan bilgisayarda kullanır.

Günümüzde, pek çok küçük işletme, bu tip program paketlerini kullanıyor. Mimarlar, doktorlar, dış hekimleri, haber ajansları, oteller, kimya laboratuvarları, petrol istasyonları, avukatlar, veterinerler, mağazalar,

lokantalar, bu tip bilgisayarları; ve dolayısıyla program paketlerini kullananlardan bazıları.

Birçok insan da, bilgisayarı çok ilginç bir hobi olarak görüyor. Bu gibi insanlar ise, bilgisayarı boş zamanlarında kullanmak üzere alırlar. Kendi kendilerine program yazmayı öğreniyorlar ve bu şekilde bilgisayarlarını ev masraflarının listesini tutmak için, yapılacak, alınacak şeyleri yazmak için, verecekleri yemekler için hazır yemek listelerini saklamak için kullanıyorlar. Ayrıca yemek tarifleri, telefon numaraları ve daha

pek çok şey bilgisayarda hem de daha düzenli bir şekilde saklanabilir.

Bir de insanın en çok ilgisini çeken oyunlar oynanabilir. Bilgisayar oyunlarının bu kadar çok ilgi görmesinin bir nedeni de, çok fazla mikrobilgisayarın piyasada olması; ve bunların en büyüğünden en küçüğüne kadar makul fiyatlarla satılmasıdır.

Şu anda, bilgisayarlara çok meraklı olan bir kişi, bilgisayarı kendi bile yapabilir. Özel olarak onun için hazırlanmış devre kartları vardı. Bu devre kartlarının üzerinde gereken



devre elemanları da vardır. Buna alan insana kalan tek şey ise, bunları gerekli yerlere koyup lehimlemektir. Bu şekilde bilgisayar hazır olur.

Konuş Benimle

BİR bilgisayarın size yazılı belgeleri sesli olarak okuması çok mu uzak geliyor? Eğer öyle düşünüyorsanız, yanılıyorsunuz. Çünkü tam da şu sıralar, sayıları pek de az olmayan birkaç şirket, bunun üzerine çalışıyor.

Video oyunlarının ilk kez yaygınlaşmasını sağlayan Uzak Oyunlardır. Çeşit çeşit oyunlar, pastanelerde, otobüs duraklarında ve daha birçok yerde ufak bir parayla çalıştırılabilir. Bu oyunlarda kullanılan bazı efektler de oyunun heyecanını artırıyor. (Solda)

Bu tip bir bilgisayar oyunuyla, bilg.sayara karşı satranç oynayabilirsiniz. Bu oyun, herhangi bir düzeydeki oyuncu için ayarlanabilir. Yeni başlayanından ustasına kadar... (Ustte)

Evde televizyonla oynamak, onu seyretmenin yanında oldukça eğlenceli olabilir. Bu oyunları ister bilgisayara karşı, ister başka bir oyuncuya karşı oynayabilirsiniz. Bellek sistemiyle de, beğenilen oyunlar saklanabiliyor, ayrıca değiştirilebiliyor da. (Altta)



Şimdilik, kendisine yazılı olarak verilen bilgileri okuma yeteneği kazanan parçalar gerçekleştiriliyor. Bu parçalar yardımıyla, bilgisayara yazılı olarak gelen belgeler, sesli olarak okunabilecek. Bu tür bilgisayarların yardımı çok fazla olacak kuşkusuz. Ama en önemli işlevleri gözleri görmeyen ya da konuşamayan insanlar için olacak. Çok sayıda şirket de bilgisayarlarla sesli haberleşme üzerine çalışıyorlar.

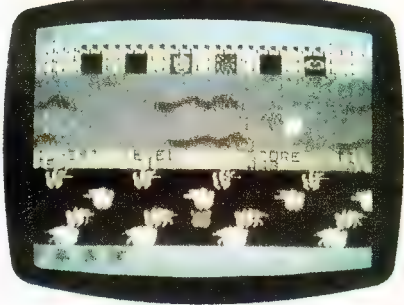
Endüstride ve askeri çalışmalarda da bu 'konuşabilen' bilgisayarlar çok işe yarayacak. Bazı makinelerin çalışmasında, kullanıcıya çeşitli komutlar verebilecek bir bilgisayar, bu makinelerin kullanımını kolaylaştıracak. Tehlikeli durumlarda da, sözlü alarm sistemi kullanılabilir. Ayrıca bu 'konuşan' bilgisayarlar, hoşça vakit geçirmemize de katkıda bulunacak. Bilgisayar oyunlarında, oyunu daha gerçekçi ve eğlenceli bir hale getirebilmek için karakterler seslendirilebilecek. Düşünebiliyor musunuz? Karşınızda konuşan bir bilgisayarla uzay oyunları oynamanın zevki bambaşka olacak herhalde. Ya da satranç oynarken, bilgisayarınız yaptığınız hamlelere yorum getirebilir. Kötü oynadığınız zaman sizinle dalga geçebilir!

Yine şu sıralar, Amerika Birleşik Devletleri'nde birçok büyük üniversite ve bilgisayar şirketi, bilgisayar sesinin insan sesine benzemesi için çalışmalarını sürdürüyor. Bu çalışmalar sonucu, insanla bilgisayar, en doğal biçimde anlaşabilecekler; günlük konuşmayla. İşte tam o zaman insan bilgisayarının yanında yalnızlık hissetmeyecek. Biraz duygunsuz olmakla birlikte bilgisayarlar, insana iyi bir arkadaş olabilecek. Hem de mantıklı düşünen bir arkadaş.

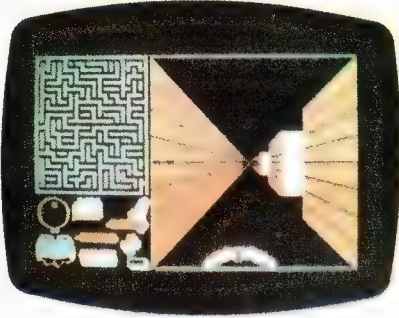
'Konuşma sentezi' üzerine çalışan bu üniversiteler ve şirketler, ilerideki yaşamımızı kolaylaştırmak amacını güdüyorlar. Konuşma sentezi, bilgisayarın günlük konuşma dilinde konuşmasını sağlamaya çalışan bir teknolojidir. Bu dil, İngilizce, Fransızca, hatta Türkçe bile ola-



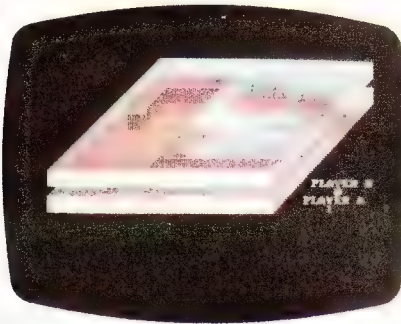
Örümcek



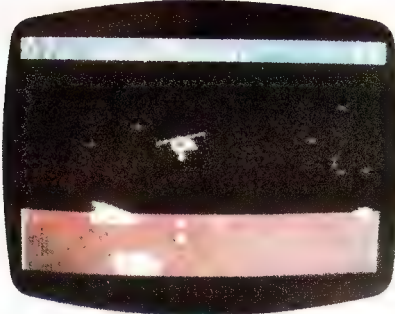
Prenses ve Kurbağa



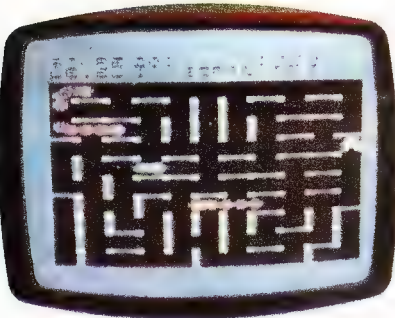
Gizli Defne



Novatran



Kaldıraç



Yılan

bilir. Konuşma sentezi, bilgisayarın konuşmasıyla ilgili diğer büyük çalışma alanından farklıdır. Bu alanda ise, bilgisayarın bizim söylediklerimizi anlamasına çalışılır. Konuşuklarımızı anlayan bir bilgisayar üzerinde de çalışmalar yapılmaktadır. Ama şimdilik pek bir sonuç elde edilememiştir. Bu teknolojiye de 'konuşma anlatımı' denir.

Bilgisayarı konuşturmak için iki önemli nokta vardır. Birincisi, analiz yoluyla sentezlemektir. Bu şekilde alet; kendisine veri olarak sunulan konuşmayı (insan sesini) alır, kelime kelime saklar ve gerektiğinde onların yardımıyla konuşur. Yani kendisine daha önceden öğretilen kelimeler yardımıyla konuşur. Bu tip konuşmayı, bir papağanın konuşmasına benzetebiliriz. O da kendisine öğretilen kelimeleri söyleyebilir. Ancak, başka türlü konuşamaz.

İkinci yöntemde ise, dilbilgisi kuralları kullanılır. Bu yöntemle "Kural sentezi" denir. Bilgisayarın belleğine kelimeler yerine heceler ve ekler verilir. Bununla birlikte tüm dilbilgisi kuralları da verilir. Bilgisayar, elindeki verilere göre konuşur. Dilbilgisi kuralları, ona kelimeleri nasıl oluşturacağı hakkında bilgiyi sağlar. Bu şekilde ona belirli kalıplar öğretmek yerine, dilbilgisi kalıplarını öğretmiş oluruz. O da yeni bir dil öğrenmeye başlamış biri gibi konuşur.

İki yöntemin de, kendine göre bazı yararları ve sakıncaları vardır. İlk yöntem, çok daha gerçekçidir. Çünkü kendisine verilen kalıpları aynen uygular. Ses de insan sesine daha çok benzer. Buna karşılık, makinenin kelime dağarcığı çok azdır. Birkaç yüz kelimeyi geçmez. Bu da bilgisayarın anlatım gücünü azaltır.

İkinci yöntemde, kelime dağarcığında herhangi bir kısıtlama yoktur. Kendisine kaydedilen dilbilgisi kurallarına uygun şekilde istediği kadar kelime türetebilir. Ancak bu makinelerin de konuşması çok makemesel oluyor. Kendine özgü bir ağızla makine, insan gibi konuşmıyor. Onun yerine, tekdüze, madensel bir ses



çıkıyor karşımıza. Bu da konuşmanın gerçekliğini bozuyor.

Son çalışmalar sonucu ortaya çıkarılan bu konuşan bilgisayarlar, daha çok donanım üzerine yapılan birtakım araştırmalar yardımıyla gerçekleştirilmiş. Bazı bilgisayarlar, ona göre hazırlanan programlarla da konuşturulabiliyor. Ama asıl işi donanım kısmı yapıyor. Bu parçalar, genelde birkaç mikroişlemci içeren kartlar ya da modüller oluyor. Doğal olarak bunlara bir güç kaynağı ve hoparlör de eklememiz gerekiyor. Bu parçaları bilgisayarımıza eklediğimiz an, bilgisayarımız konuşmaya başlı-

yor. Ama, bilgisayardan ayrı bir parça kullanmak daha yaygın durumda. Ufak bir yazıcı büyüklüğündeki parça, kendi güç kaynağını ve hoparlörünü kendi içinde taşıyor. Bu parça, bilgisayara dışarıdan bağlanarak konuşmasını sağlıyor.

Bu parçalar, şu anda dünya piyasasına sunulmuş durumda. Değişik fiyatlarda değişik parçalar, bilgisayarınıza bağlanarak onun konuşmasını sağlıyor. Hepsi aşağı yukarı aynı şeyi yapıyor. Kendisine yazılı olarak verilen bilgiyi, yine kendisine daha önceden verilen kalıplara uygun olarak ya da dilbilgisi kuralları

yardımla okuyor ve sesli olarak kullanıcıya aktarıyor. Yazılı veriler de kullanıcının isteğine göre harf harf, kelime kelime, satır satır'ya da tümünden okuyulabiliyor.

Eğer parçaların hepsi aynı işi yapıyorsa, aralarında niçin fiyat farkı var diyebilirsiniz. Öyle ki, bir parçanın fiyatı 16 katı olabiliyor. Bunun nedeni de, içinde kullanılan parçaların kalitesine bağlıyor. Bir parçanın üzerinde ne kadar çok çalışılmışsa, ne kadar karmaşık devreler kullanılmışsa, bu onun kalitesini, dolayısıyla fiyatını etkiliyor. Konuşma yeteneği daha gelişkin

olan parçalar, daha pahalıya satılıyor.

Ancak parçaların tutarı arasındaki fark, kalitelerinde iyice belirginleşiyor. Yani daha iyi kalitede olan parçalar, diğerlerinden hemen ayrılıyor. Ucuz parçalarda kullanılan ses, kesinlikle insan sesine benzemiyor. En geniş hayal gücü olanlar bile, bu görüşü paylaşmak zorunda kalıyorlar. Bu ses, daha çok erkek sesini andırıyor. Ancak vurgular ve ağız çok farklı. Sözcüklerde de hiçbir anlam bulunmuyor. Tekdüze, makineden geldiği kolayca anlaşılacak bir ses kullanılmış. Ayrıca kullanıcının söylenilenleri kelimesi kelimesine büyük bir dikkatle izlemesi gerekiyor. Çünkü her an aradan bir sözcük kaçırma olası. Aradan bazı heceler atlanabiliyor, bazı söz-

cükler yutulabiliyor. Özellikle kullanıcı tarafından pek bilinmeyen sözcükler yanlış anlaşılabilir. Bu da doğrulukla kullanımı zorlaştırıyor.

Bir de diğer tip parçalara bakalım. Bunlar büyük bir dikkatle hazırlanmıştır. Ses, hemen hemen insan sesine benzer. Hatta konuşmasında bir aksan bile bulabiliriz. Bu da uzun uzun hazırlanan çiplerle gerçekleştirilebilir.

Bu teknoloji oldukça pahalıya mal oluyor. Çünkü hem geniş bir bellek kapasitesine, hem de hızlı çalışabilme gücüne sahip olması gerekiyor. Konuşma sentezi yönteminde amaç, en gerçekçi sesi, en ekonomik bir biçimde yakalayabilmek. Konuşma, bellekte çok fazla yer kaplıyor. Normal bir hızda yapı-

lan bir konuşma da saniyede 8K'lık bir hızla gidiyor. Bu en fazla 2,5K'ya kadar indirilebilir. Ayrıca bu tip bir çalışma için disk kullanılmadığı zaman 64K'lık bir bellek saniyede 2,5K'lık bir hızla 25 saniye için kullanılabiliyor. Bu da yalnızca 90 sözcük ediyor. Yani bu şekilde ancak 90 kelime-lik bir yazıyı bize sesli olarak okuyabiliyor.

Konuşma sentezi ile hazırlanan belleklerde, sözcüklerin bir bütün olarak verilmesi gerekiyor. Bu da onların sözcük kapasitesini alabildiğine azaltıyor. Bu nedenle, konuşma sentezi yerine birçok yapımcı kural sentezi yöntemini kullanmayı yeğliyor. Bu şekilde bilgisayar, kendisine verilen yazıları fonetik kurallarına göre okuyor. Tıpkı yeni bir dili öğrenenler gibi elinde okuma kuralları, iste-



nileni okuyor. Bu yöntemle doğaldır ki, sözcük kısıtlaması ortadan kalkıyor. Eğer gerekli kurallar elindeyse, makine istenileni okuyabiliyor. Hem de anlaşılabilir bir şekilde...

Bu yöntemin kullanımında da aşılması gereken iki problem var. Bunlara problemlerden çok, altından kalkılması gereken sorunlar da diyebiliriz. Birincisi fonetik, yani okuma, kurallarını bilgisayarın anlayabileceği bir biçime getirmek; ikincisi de bilgisayarın kurallara uygun bir şekilde ve gerçeğe yakın bir sesle konuşmasını sağlamak. Bu arada yargulamasına ve kelimeleri tane tane okumasına da dikkat etmek. Bu iki problemin çözümü, birbiri ardından gerçekleştiriliyor.

Şu anda, bilgisayarın konuşması üzerine çalışan şirketler Amerika Birleşik Devletleri'nde olduğu için, bilgisayarlar yalnızca İngilizceyi konuşabiliyorlar. Bu dilin de fonetik kuralları oldukça karışıktır. Sözcükler, yazıldığı gibi okunmaz. Bu yüzden de oldukça farklı okunan sözcükler vardır. Bunların bilgisayara verilen bilgiler arasında belirtilmesi gerekir. Ayrıca konuşmaya normal akış hızını kazandırmak gerekir. Bu hızın bilgisayara belirtilmesi zorunludur. Bir de bütün bunlara ek olarak, okunan yazılara az da olsa anlam kazandıracak bir tonlama verilmelidir. Düz cümlelerin okunması, soru cümlelerinin okunmasından farklı olmalıdır. Bu da bilgisayara apayrı bir bilgi gerektirir.

Bilgisayarı konuşturmak için temelde iki yapılan şey, onlara tüm değişik sesleri öğretmektir. Buna bir örnek verelim. Kapsöz-cüğünde dört değişik ses vardır: k, a, p ve ı sesleri. İlk olarak, bu seslerin tümü bilgisayara verilir. Ancak iş bununla bitmez. Yani yalnızca bu seslerin yardımıyla bilgisayar sözcükleri hiçbir şekilde doğru olarak okuyamaz. Çünkü bu sesler, her sözcük içinde değişik biçimlerde söylenebilir. Bu da, sözcük içinde bulunan diğer seslerin biçimine göre değişir. Bunun için, değişik ses



biçimlerine göre, değişik sesleri de vermemiz gerekir. Yani değişik şekillerde okunan hece tipleri vermeliyiz. Yalnız iş bununla da bitmiyor. Bu hecelerin gerçeğe uygun bir biçimde bitleştirilip söylenilmesi gerekiyor. Sözcüklerde vurgunun hangi hece üzerinde olacağı, yapılan tamlamalara göre bu vurgunun nereye kayacağı belirtilmeli. Ayrıca cümlelerin soru cümlesi mi, yoksa düzcümle mi olduğu anlaşılmalı ve ona göre cümlede değişik tonlamalar yapılmalı. Noktalama işaretlerine dikkat edilmeli. virgüllerde bir parça, noktalarda biraz daha fazla durulmalı.

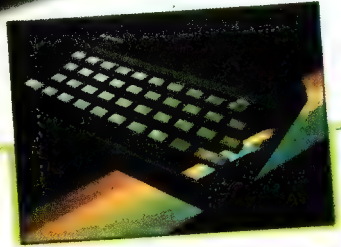
Bütün bu noktalara dikkat edilmedikçe, bilgisayarın güzel bir şekilde konuşması düşünülemez. Yine de yukarıda saydığımız koşullar, bilgisayarın kusursuz konuşmasını sağlamaya yetmiyor. Bir cümlemin tam anlamıyla doğru bir biçimde okunabilmesi için, onun anlaşılması gerekiyor. Bu da şimdilik bilgisayar için oldukça güç bir şey.

Evet, bilgisayarlar artık konuşabiliyor. Ama henüz tam

gerçekçi bir biçimde değil. Bir süre için kendilerine has aksanlarını koruyacaklar. Yapımcılar, önderindeki sorunları biliyorlar, ama henüz bilgisayar üzerinde bir çözüm bulamıyorlar. Bu sorunların da bilgisayar alanında çözümü bulunduğu zaman, bilgisayarlar da bir insan gibi konuşabilecekler.

Şimdi asıl işleme geçelim. Bilgisayarın konuşabilmesi için gereken kurallar saptandıktan sonra, bunların yazılım ya da donanım bölümünde gerçekleştirilebileceğinin saptanması gerekir. Bu kurallar hem programlarda, hem de devrelerde gereken şekliyle belirtilir. Sonra da bunları insan sesi haline dönüştürecek bir alet gerçekleştirilir. Biz bu aleti bir şekilde radyoya benzetebiliriz. Kendisine gönderilen sinyalleri, insan sesine benzeterek sözcüklerle dışarıya gönderir.

Konuşan bilgisayarların bize saylayacağı yararlar kuşkusuz saymakla bitmez. Ama şu anda yapımcılar için çok önemli görünen birkaç nokta var.



Bunlar, öncelikle elde edilecek kolaylıklar. Doğaldır ki, bunları diğerlerinin de izleyeceğini söylememize gerek yok.

İş hayatında bilgisayarların konuşarak sağlayacağı kolaylıkları şu şekilde özetleyebiliriz. Birincisi, yazılı olarak saklanan dökümanların önceden okuma işlemini yapması. Pek çok yazılı belgenin bu şekilde okunması hem çalışanların zamandan tasarruf etmelerini sağlayacak, hem de bazı noktaların okurken gözden kaçmasını önleyecek.

İkinci önemli nokta, özellikle bilgisayar kullanımını kolaylaştıracak. Bilgisayar kullanımını yeni öğrenen bir kişinin yanlış yapma olasılığı çok yüksektir. Ya bu kişiler deneyim kazanana kadar bir kursa gönderilirler, ya da ellerine makineyi tanıtıcı birkaç yazı verilir. Kurs, çok iyi bir

yöntemdir. Ama her zaman olmayabilir. Buna karşın kullanıcının elinde bulunan yazılar da ona pek yarar sağlamaz. O, yanlış yaptığı zaman kendisine yol gösterecek bir şey ister yanında. Bunun için konuşan bir bilgisayar, onun çok işine yarayabilir. Kullanıcı zor durumda kaldığı zaman bilgisayardan yardım isteyebilir. Bu, şimdi kullanılan bilgisayarlarda da yapılan bir şeydir. Ama yardım için ekranın kullanılması gerekmektedir. Bu da işi yavaşlatır. Onun için konuşan bir bilgisayar ekranı meşgul etmez ve kullanıcıyı zor durumdan kurtarır.

Üçüncü nokta, bu tip bilgisayarların haberleşmede kullanıla-

bilmesi. Bunun için kullanıcının sürekli olarak yanında konuşan parçayı taşıması yeterli. Böylece bilgisayarına gelen notlar, doğrudan kendisine okunmuş oluyor.

Konuşan bilgisayarların iş hayatından başka yerlerde de büyük yardım sağlayacağı kuşkusuz. Gözleri görmeyen ve konuşmayan insanlar için bu konuşan bilgisayarlar bir umut ışığı oluyor. Bu bilgisayarlar yardımıyla artık görmeyen ve konuşamayan insanlar da bilgisayar kullanabilecek ve diğer insanlarla iletişim kurabilecek.

Bir yapımcı, gözleri görmeyen insanlar için özel bir bilgisayar hazırlamış durumda. Bu

bilgisayar, klavyesinde basılan her tuşu anında sözlü olarak belirtiyor. Bu şekilde bilgisayarı kullanan kişi, gözleri görmese de hangi tuşa bastığını hemen öğrenebiliyor ve bu şekilde yanlışları da düzeltebiliyor. Yani konuşan bilgisayarların kendilerine verdiklerini sözlü olarak duyabilecekler. Yine aynı kişiler, kendilerine yazılı olarak gönderilen notları da aynı şekilde okuyabilecekler. Konuşmayan insanların da bir anlamda konuşması sağlanacak. Belki bir gün bu çalışmalar öyle bir aşamaya gelir ki, bu konuşmayan insanların içine konuşmalarını sağlayacak bir elektronik parça takılabilir.

Ayrıca, daha önce de belirttiğimiz gibi bu konuşan bilgisayarlar, askeri işlerde de büyük kolaylık sağlayacaklar. Herhangi bir alarm sistemi sözlü olarak yapılsa, hem daha geniş bir alanda kullanılabilir, hem de daha etkileyici olur.

Şimdi de üzerinde çalışılan nokta, bilgisayarların günlük konuşma dilinden anlamasını sağlamaktır. Yakın bir gelecekte bu da gerçek olacaktır. Bu şekilde biz de, değişik bilgisayar dillerine eskisi kadar gereksinim duymayacağız ve bilgisayarlar iyice günlük yaşamımıza girmiş olacaklar.

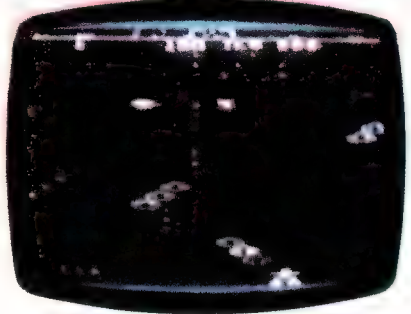
Bilgisayara Görmeyi Öğretmek

BİLGİSAYAR yapımcıları, bilgisayarın yaratıcı işlerde de kullanılabileceğini söylüyorlar. Bilgisayar, kullanışlı olduğu kadar eğlenceli de olmalı.

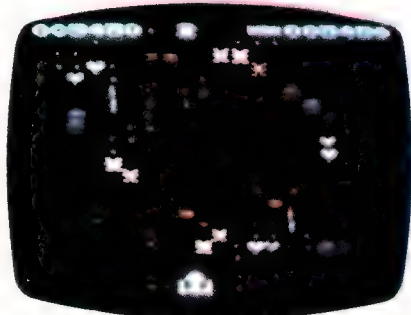
Bilgisayarın bu tip işlerde kullanılması iyice yaygınlaştı. Ancak yirmi, yirmi beş yıl öncesine kadar, bu görüşü paylaşılanların sayısı pek de fazla değildi. O



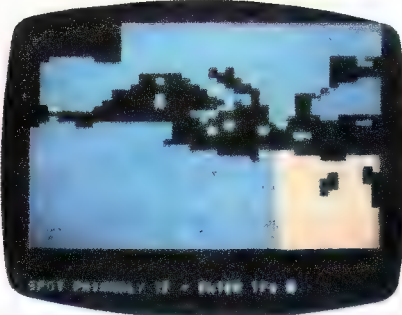
1985 Yılı



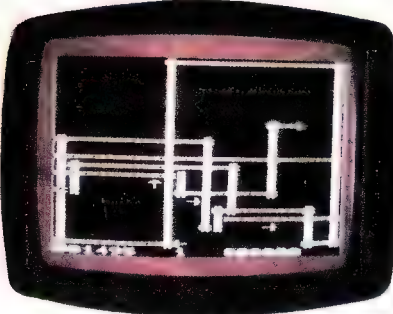
Astroblitz



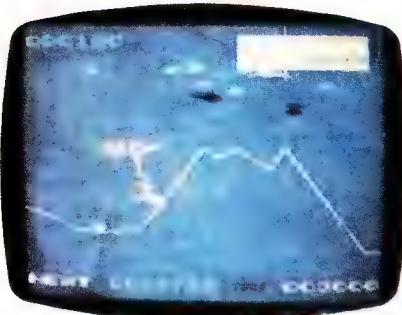
Bataklık



Bomba Tuzağı



Köpekbalığı Tuzağı



Füze Oyunu

zamanlar, bilgisayarlara gelişmiş birer hesap makinesi gözüyle bakılıyordu. Onlar, çok ciddi işler için kullanılmalydılar. Onlarla hoşça vakit geçirilmezdi.

Son yirmi yıl içinde, birçok bilgisayar oyunu programlandı. Bunlar pek çok bilgisayar diline çevrildi ve bu şekilde değişik bilgisayarlarda kullanılması sağlandı.

Bu oyun programları, çıkan bilgisayar dergileri aracılığıyla diğer kullanıcılara da ulaştı. Bu şekilde bilgisayarların yalnızca ciddi işlerde değil, eğlenceli işlerde de kullanılabileceği kanıtlanmış oldu.

Bundan sonra, bilgisayarla başka denemelere girişildi. Bilgisayarların içinde, kimi zaman bize gerekmeyecek kadar fazla bilgi oluyor. Bizim bu gerekmeyenleri ayıklayıp çıkartmamız gerekiyor. Ya da bazı bilgiler, o kadar uzun ve karmaşık oluyor ki, anlamamız epey vakit alıyor. Bunun için bambaşka bir yol düşünüldü.

Bu sayıların grafiklerle anlatılması çok daha kolaydı. Ayrıca bir resim binlerce sayının, yüzlerce kelimenin yerini tutabiliyordu. Bu göz önüne alınarak, bilgisayarlara grafik çizdiren programlar yapılmaya başlandı. Bugün, kusursuz biçimde çizilmiş bir grafik, bizi bütün bu karışıklıklardan koruyor.

Bu grafik, Türkiye'nin coğrafya haritası olabilir. Nüfus yoğunluğunu gösteren bir harita olabilir. Ya da bir denklemin grafiği olabilir. Ya da bir şirketin gelir giderlerini gösterebilir.

Bu resimler, hareketli de olabilir. Yani alana inmek üzere olan bir uçağın görüntüsü olabilir. Bu tip bir resim, işe yeni başlayan bir pilota çok yardımcı olabilir.

Pilotun, bilgisayarı kullanabilmesi için uçağa binmesi gerekmiyor. Yalnızca ekranın önünde



uçacağı indirip kaldırma çalışmalarını yapıyor. Bu şekilde pilotun uçacağı düşürme korkusu olmadan deneyim kazanması sağlanabiliyor.

Bilgisayarın Gözüyle Dünya

YİNE bir süre öncesine kadar bilgisayar ekranları, bilgisayarın programlarını ve bilgilerini göstermeye yarayan bir alettir. Yani bilgisayarlar bir hesap aletiydi ve ekranlarının da hesapları göstermekten başka bir işe yaraması düşünülemezdi.

Ama şimdi o ekranlarda, renkli ya da siyah beyaz isimler yapılıyor. Bu resimler o kadar

gerçekçi oluyor ki, fotoğraf izlenimi veriyor.

Bu resimler, uzun, karışık bir dizi programın sonucunda ortaya çıkıyor. Eğer resmi beğenmezseniz, bir komutla yenisini çıkarıyorsunuz.

Evet, herkes bilgisayarın aritmetikte iyi olduğunu biliyordu. Ama kimse onun resim de yapabileceğini düşünmemişti.

İlk Elektronik Resim

BİR gün, bir programcı bilgisayarla oynarken, ona çizgi çizdirmeyi düşündü. Bugüne kadar hiç kimse, bilgisaya-

yara resim çizdirmek için bir program yapmayı düşünmemişti. Bu tip çizgiler, bilgisayarın klavyesindeki tire tuşuna sürekli basılarak elde ediliyordu.

Bizim programcı, bilgisayara uzun bir çizgi çizdirmek istedi. Ancak önünde, aşılması gereken bir dolu problem vardı. Bu çizgiyi çizdirmek için hangi komutu kullanmak gerekiyordu? Ona nereden başlayıp nerede bitireceği nasıl söylenecekti? Ayrıca bu çizgi sabit bir uzunlukta mı olacaktı?

Bunun yanıtını bulmak altı aydan fazla zaman aldı. Ama sonunda bilgisayara bir doğru çizdirmenin yolu bulunmuştu! Bundan sonra programcımız, karenin ve üçgenin nasıl çizildi-



ğini çıkardı. Ve en önemlisi, dairinin nasıl çizildiği bulundu.

Ama programcımız burada bulamadı. Sıra üç boyutlu figürlere gelmişti. Küreler, üçgen prizmalar, kolaylıkla çizilebiliyordu artık. Sonra bu figürler, kendi çevrelerinde dönmeye başladılar. Sonra da onlara büyüüp küçülmeleri öğretildi. Daha doğrusu bu tipte programlar hazırlandı.

Bu alanda adım adım gelişme sağlanıyordu artık. Yavaş yavaş bilgisayar resimlerine geçildi. Doğaldır ki, bu iş, programcılarının iki buçuk yıldan fazlasını aldı. Ama değerdirdi doğrusu!

Ancak, bu resimler karşılarnı yeni bir tip problem çıkardı-

lar. Ne kadar güzel olurlarsa olsunlar, onları kimse alıyordu. Bunun nedeni de, kimsenin bu resimler hakkında bilgi sahibi olmamasıydı. Onların ne gibi işlerde kullanılabileceği de bilinmiyordu.

Bir resmin binlerce sayıya, yüzlerce kelimeye bedel olduğu anlatıldı herkese. Bunu da başarmak çok zordu. Çünkü, herkes bilgisayarı bir hesap makinesi olarak görmeye devam ediyordu. Onun için resme ne gerek vardı?

Bütün bu sorulara birer yanıt bulunması gerekiyordu. Bulundu da. Sonunda bu resimlerden endüstride birtakım yararlar sağlanabileceği anlaşıldı. Şimdi birçok kuruluş, bu resimlerden yararlanıyor. Bu resimlerle grafikler çiziliyor, dersler anlatılıyor. Yakında belki çok daha geniş bir alanda kullanılacaklar.

Alfabe Ressamları

BİR başka programcı ise müzisyendi. Bazı insanlar, en iyi programların müzisyenlerden çıktığını söylerler. Belki bu bir bakıma doğrudur da. Çünkü bu müzisyen de aynı zamanda çok iyi bir programcıydı. Bir üniversitede bilgisayar dalında öğretim görevlisi olarak çalışıyordu.

Bu programcı da, kullanılan harflerin ve sayıların değişik puntolarda yazılması üzerine araştırmalar yapıyordu. Doğallıkla bu işi de bilgisayara yaptırmak gerekiyordu. O sıralar yeni yeni kullanılmaya başlanan fotoğrafik basım metodu, programcının bir öğretim görevlisi olarak yazdığı kitapların kalitesini düşüreceği görüldü. İşte bu nedenle bilgisayara yazı yazdırma işi öğretilmesi gerekiyordu.

Bunun için, önce bu harfler ve rakamlar için özel programlar hazırlanması gerekiyordu. Doğru, daire, üçgen ve diğer geomet-

rik şekillerin programlarından yararlanarak, bu harflerin ve rakamların programlarını buldu. Bu programlar yardımıyla istenilen büyüklükte yazılan yazılar, istenilen yöne doğru kaydırılabiliyordu, büyüüp küçülebiliyordu.

Sonra bu yazılar, aynen yazıcıya da aktarılabilirdi. Yani yazıcı, ekranda görülen yazıları aynen kaydedip çıkarabiliyordu.

Bu şekilde, programcımız, kitaplarını istediği gibi basabili. Ayrıca kendi yarattığı bir program dilini de satışa çıkarttı. Bu program yardımıyla artık herkes istediği biçimde yazılar yazabilecek, kitap basabilecekti.

Elektronik Kitap

BU gelecekte herhangi bir gün başınıza gelebilir. Odanızda ders çalıştığınızı düşünün. Ertesi güne bir tarih sınavınız var. Diyelim ki, Kurtuluş Savaşındaki zaferlerin tarihlerini kavramaya çalışıyorsunuz.

Ancak pek verimli çalışmıyorsunuz. Çünkü daha önce çok yorucu bir iş yapmışsınız ve onun yorgunluğunu daha üzerinizden atamamışsınız.

Nereden geldiği pek belli olmayan bir ses, siz tam gözlerinizi kapadığınız bir sırada sizi şaşkına çeviriyor:

— **Büyük Taarruzun tarihini söyle!**

— Efendim?

Siz ancak gözlerinizi açıp konuşabiliyorsunuz.

— **Büyük Taarruzun tarihini söyle.**

— Kitabınızı açıyorsunuz. Ama sayfaların üzerindeki yazılar yok olmuş. Onun yerine koskocaman bir yazı var:

Büyük Taarruzun tarihi?
Onu ezberlemiştiniz. "30 Ağustos 1922" diye yanıtlıyorsunuz.

Karşıdan da yanıt geliyor.
— **Doğru. Bildin.**
Ses, kitabınızdan geliyor.



Kitabınız size ders çalıştırıyor. Sizinle hem konuşuyor, hem de sorularını ekrana bir kez daha yazıyor. Birazdan ekranınız kararıyor ve yeni bir soru geliyor karşınıza:

—İzmir'in kurtuluşu tarihini söyle.

Biraz zorlanıyorsunuz. Bir türlü aklınıza gelmiyor. Biraz düşünüyorsunuz.

— 4 Ekim 1922'miydi?

— Hayır, yanlış anımsıyorsunuz.

Bir kez daha düşün.

— 9 Eylül 1922'miydi?

— Evet, bu kez bildin, aferim.

Şimdi de...

Bu böyle sürüp gidiyor. Bir süre sonra bir de bakıyorsunuz, bütün tarihleri ezberlemişsiniz. İşte size çalışan ve çalıştırın bir makine. Yakında bütün öğrencilerin odalarında bu tip bir bilgisayar olabilir.

Bu anlatıklarımız şimdilik size biraz uzak gelebilir. Kitap

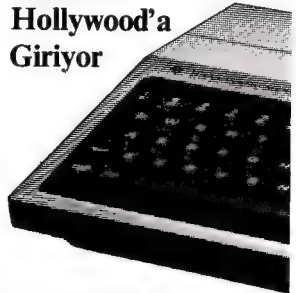
şeklinde bilgisayar nasıl olabilir diye düşünebilirsiniz. O kâğıtların içine hiçbir çip sığmaz, diyebilirsiniz. Ama yakın bir gelecekte, diyelim ki on yıl sonra, bunlar gerçek olabilir. Doğaldır ki, kitaplarımız içine bilgisayar yerleştirilemez bugünkü haliyle. Ama onları kâğıttan değil de plastik kartlardan yaparsak, pekâlâ da bilgisayarlı olabilirsiniz. Sayfayı bitirdiğiniz an bir düğmeye basıp bir sonraki sayfaya geçebilirsiniz.

Eğer okumaktan yorulursanız ya da okumaya zamanınız yoksa, kitabın başka bir düğmesine basarsınız, o size okur. Gözlerinizi kapatıp onu dinleyebilirsiniz.

Kitap size sayfaları gösterip okumakla da kalmayacak. Resimler de gösterecek. İçine yerleştirilen programlarla size sözlü yapabilecek, alıştırmaları soruları verebilecek. Bu kitapların yanın-

da sürekli tetikte olmanız gerekecek. Çünkü siz tam uyumaya başlarken, size zor bir soru sorabilir.

Bilgisayar Hollywood'a Giriyor



WALT Disney'in Pinokyo' sunu hiç gördünüz mü? Dev balina Monstro'nun Pinokyo'yu nasıl yuttuğunu, onu

nasıl içersinde hapis tuttuğunu anımsarsınız herhalde.

Ya da Miki Fare'yi ve diğer kahramanların filmlerini. Bütün bu çizgi filmler elle yapılır. Her resim tek tek elle hazırlanıyor. Bu filmler çok gerçekçi bir biçimde hazırlanmıştır. Karakterler canlı, akıllıdır. Hareketlidir. Renklidir. Ancak bu filmler, bir süre sonra bu çizgi filmler eskisi kadar çok yapılmamaya başladı. Çünkü yapımı zor pahalıya maloluyordu. Çünkü kısıcak bir çizgi film için bile binlerce resme gerek vardı. Onların tek tek hazırlanması, boyanması gerekiyordu.

Eğer az resim kullanılırsa, filmin kalitesi düşüyordu. Karakterlerin hareketleri kabalaşıyor, sertleşiyordu. Bir dudak hareketlerini göstermek için bile yüzlerce resim çizmek gerekiyordu.

1980'li yıllarda bilgisayar kullanılmaya başlandı. Bu şekilde yeniden seri halinde çizgi film yapılmaya başlandı.

Bilgisayarlar, çizgi film piyasasına yeniden canlılık getirdiler. Doğaldır ki, her işi onlar yapıyordu. Ana fikri yine ressamlar veriyor, onlar çiziyordu. Bilgisayarın iki önemli işlevi var. Birin-

tekdüze işleri yapmaktan kurtarıyor. Bu işler arasında resimleri renklendirme ve aynı fonu tekrar tekrar çizmek var.

Ressam, ana resmi çizer ve bilgisayarın belleğine yerleştirir. Bilgisayara, sürekli ona bu resmi çıkarır. Sonra da, hangi parçanın ne renk olacağını bilgisayar gösterir. Bilgisayar da onları otomatik olarak renklendirir.

Örneğin ressam bir robot çizmiş ve onun metalik mavi renkte olmasını istiyor. Bilgisayara istediği koordinatları verdiğinde ya da özel bir kalemle robotun üze-



cisi, ressamın çabuk iş görmesini sağlıyor. İkincisi de onu sıkı,

rine dokunduğunda ve klavyede de mavi tuşuna bastığında, robot otomatikman mavi olur. Doğallıkla bu bilgisayarlar bir iş için özel hazırlanmıştır.

Örneğin bu robotun bir yenden atlaması gerekiyor. Eğer bilgisayar kullanılmamış olsaydı, ressamın yüzlerce resim çizmesi gerekirdi. Bu resimlerde robot hariç her şey hareketsiz kalıyor. Ve ortaya ressamın çizmesi gereken yüzlerce aynı tipte resim kalıyor. Ayrıca resmin gerçekçi olabilmesi için aradaki hareketleri göstermek gerekiyor. Bu arada hareketler ne kadar fazla olursa, film de o kadar gerçekçi olur.

Artık bu iş için ressamın o



kadar resim çizmesine gerek kalmıyor. Onun yerine iki resim çiziyor. Bunlardan biri robotun ilk durumunu, biri de son durumunu gösteriyor. Bilgisayar da otomatik olarak, kendisine verilen bu resimlerin arasında bulunması gereken resimleri çizebiliyor.

Bu şekilde, bilgisayar yardımıyla çizgi filmler, yeniden altın çağına geliyorlar. Çok ucuza üretilen bu filmler, artık her yerde boy gösteriyor.

Dördüncü Boyuta Doğru Bir Pencere

BİLGİSAYARLARA çok güzel resimler çizmeyi öğretebiliriz. Bu resimlerle istediğimiz şeyleri gösterebiliriz. Aynı şekilde bu ekranı gizemli dördüncü boyuta giden bir yol olarak da düşünebiliriz.

İlk boyut uzunluktur. Bir doğru, bu boyutu gösterir. Doğrunun uzunluğu vardır. Ama başka hiçbir şeyi yoktur.

İkinci boyut genişliktir. Kare, iki boyutlu bir şekildir. Uzunluğu ve genişliği vardır, ama düzdür.

Üçüncü boyut da yüksekliktir. Hepimiz üç boyutlu yaratıklarız. Hepimizin uzunluğu, genişliği ve yüksekliği var.

Çevremizdeki dünya, üç boyutlu gibi görünür. Ağaçlar, bulutlar, arabalar ve aklınıza gelebilecek her şey üç boyutludur.

Peki dördüncü boyut nedir? Hayaletlerin, cinlerin, perilerin evi mi?

İlkçağ düşünürleri, dördüncü boyutun olmadığını iddia ediyorlardı. 1600 yıllarında ortaya çıkan bir matematikçi ise, dördüncü boyutun doğadaki canavarlar olduğunu ileri sürdü...

Ama yirminci yüzyıl başlarında, Albert Einstein dördüncü boyutu gerçek yerine koydu. Dördüncü boyut zamandır. İlk üç boyut, bu dördüncü boyutun içerisinde sürüklenip dururlar.



Zamanın geçtiğini anlayabiliriz. Ancak diğer boyutlar gibi gözümüzle göremeyiz. Başka bir şekilde, dördüncü boyut görünmez bir şeydir. Ama Einstein, bu teorisinin üzerine birçok denklem kurmuş ve evreni anlamanın tek yolunun bu dördüncü boyutu anlamak olduğunu belirtmiştir.

Kaybolan Boyut

SÖYLE bir örnek verelim. Örneğin siz yolda yürüyorsunuz. Bu da filme çekiliyor. Bu oldukça doğal bir şey, değil mi? Ama üzerine güneş vurdu diyelim. Güneş sizin gölgenizi çıkaracaktır. Gölge, sizin gölgenizdir, bilirsiniz. Yani sizin vücudunuzu gösterir. Ama iki boyutludur. Oysa siz üç boyutlusunuz. Ne yaparsanız yapın, gölgenizi üç boyuta çıkaramazsınız.

Zaman da böyle bir şeydir işte. Güneşe benzer. O da üç boyutlu olarak yansır. Onu dördüncü bir boyut olarak çizmemiz olanaksızdır, ama üç boyutlu gölgelerini çizebiliriz. Bu gölgeleri de bilgisayara çizmeyi öğretebiliriz.

1970'li yılların sonunda, iki araştırmacı, bunun üzerine yapı-

lan çalışmalarını bir sonuca bağlamayı başarmıştı. Bu iki araştırmacı, 'hiberküp' adı verilen bir şekli bilgisayara çizdirme işini gerçekleştirdiler. 'Hiberküp' adı verilen bu şeklin gerçek yaşamda gerçekleştirilmesi olanaksızdı. Çünkü dört boyutluydu. Bilgisayara da ancak gölgesi yansıtılabiliyordu.

Bu gölgeler şu şekilde gerçekleştiriliyordu. Gökkuşağına benzer bir şekil, ekranda durup dinlenmeden dönmeye başlıyordu. Renkler dönüp durdukça, seyredenlerin karşısına lunaparklardaki garip aynaların başına geçmiş gibi garip şekiller çıkıyordu. Şekil sanki sürekli kendi içinde hareket ediyor, içini dışına çeviriyordu. Dördüncü boyutun yansıtılması bu şekilde gerçekleştirildi.

Müzik Makinesi

ÜZİK kıyıda kenara çarpan dalgalara benzetilebilir. Yalnız müziğin dalgaları suda değil havada olur.

Eğer bir trampetin çıkardığı ses dalgasını, ya da bir gitarinkini çizmek isterseniz, karşınıza denizdeki dalgalara benzer bir şekil çıkar.

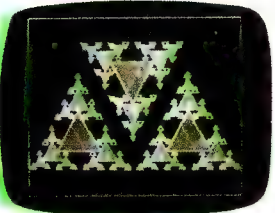
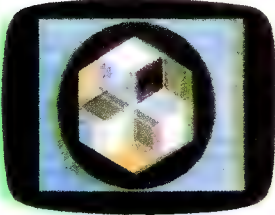
Bilgisayara bu dalgayı çizdirmek isterseniz dalga üzerine ufak merdivenler çizeriz gerekir. Bu merdivenler dalgayı izleyerek önce yukarı çıkar, sonra da aşağıya iner.

Bilgisayar, ses dalgalarını üretmek için işe elektrik uyarılarını göndermekle başlar. Dalga boyunca çizdiğiniz o basamaklardan her biri, bir uyarı olur. Onlar da dalga yukarı çıkıkça çıkar, indikçe inmeye başlar.

Hızlı bir bilgisayar; tek bir ses dalgası çıkarabilmek için düzinelerle uyarı kullanabilir. İlk uyarılar, dalganın başlangıcında olduğu gibi kısadır, küçüktür. Ortadaki uyarılar büyüktür. Bunlar da dalganın tepesini gösterirler. Yine son uyarılar da, dalganın sonu gibi küçüktür.

Bilgisayar, bu uyarıları birtakım bağlantılar aracılığıyla hoparlöre gönderir. Hoparlörde, bu uyarıların yarattığı titreşimler, havaya ses dalgaları olarak yansır.

Havada hareket eden ses dalgaları, sizin kulağınıza kadar



gelir. Kulağınızdan içeri girip kulak zarınızda titreşimler yaratır. Bu titreşimler de beyne gönderilir. Beyinde de ses olarak algılanır. Yani sonuçta biz müziği duymuş oluruz.

Bilgisayarı bir müzik aleti olarak kullanan ilk kişi bir öğrenci olmuş. Bu öğrenci, önceleri geliştirdiği bir sistemle bilgisayarın gönderdiği elektrik uyarılarını kontrolü altına almış. Onları yerleştirdiği düğmeleri çevirerek uzatıp kısaltmaya başlamış. Böylelikle istediği tonda sesleri çıkartmayı başarmıştı.

Ancak bir süre sonra bu yöntemden sıkılmaya başlamış. Yani durmadan düğmeleri çevirmekten yorulmuş. Uzun çalışmalar sonucu yeni bir yöntem bulmuş. Uzun kâğıt şeritleri üzerine, kendi geliştirdiği kodlama siste-

miyle notaları yazmaya başlamış. Bu notaların yazımı, bilgisayarın anlayacağı bir dilde, yani kart delgi sistemiyle uygun bir biçimde gerçekleşmiş. Açılan deliklerin biçimi, değişik notaların kodlanması için kullanılmış.

Bu şekilde, bilgisayar okuduğu her notanın ses dalgalarını üretemeye başlamış. Yani bilgisayarı, bir tür elektronik piyano haline gelmiş.

Son araştırmalar sonucu gerçekleştirilen bir bilgisayar ise, yalnızca bu tür işleri için kullanılıyor. Özel programlar yardımıyla bu bilgisayar, sevilen müzikleri çaldığı gibi, bestecilere de yardımcı oluyor. Kendisine verilen notaları belleğine kaydediyor, istenildiğinde baştan çalıyor.

İşte, bilgisayar artık çok değişik yerlerde ve çok değişik amaçlarla kullanılabiliyor.



1. Yolcu, Los Angeles'ten uçağa binmek üzere bilet istiyor. Uçuş hakkında ayrıntılı bilgi öğrenebilmek için terminale başvuruluyor.

2. Ana bilgisayar, otomatik olarak istenen bilgileri Londra bürosunun ekranında gösteriyor.

Havayolları
bürosu (Londra)

Uydu bağlantısı



3. Yolcunun bilet kaydı, Chicago'daki bilgisayara gönderilmek üzere yazılıyor

Houston'daki Amerikan Apollo uzay araştırmalarını yönlendiren kontrol merkezi. Yeryüzünde, ana uzay araçlarında bulunan bilgisayarların yardımı olmasaydı, belki de bu geziler gerçekleşmeyecekti. Astronotlar da, bilgisayarların yardımıyla eğitilmişlerdir. (Üstte)

yayılmış olabilir. Bu ünitelerden bazıları "akıllı terminal" denilen türdensen, bunlar birtakım hesaplamaları kendi içlerinde yapabilir. Yani bu tip bilgisayarlar bazı işleri ana üniteye bağlanmaksızın gerçekleştirebilirler.

Güvenlik İşleri: Çoğu güvenlik bürolarında bilgisayarlar için çok iş vardır. Acil durumlarda zamanında önlem alabilmek için kullanılırlar. Büyük şehirlerde ise trafik akışını düzenli tutabilmek için trafik lambaları, bilgisayarlar tarafından kontrol edilir.

Güvenlik örgütlerinde suç kaydı olan insanlar hakkında toplanan bilgiler, parmak izleriyle birlikte bilgisayarda saklanır. Bu, aranan suçluları bulmakta polise büyük yardım sağlar, aynı tipte bir suçu daha önce işlemiş insanların kayıtlarından yararlanılabilir.

Bilgisayarlarda toplanıp, kolayca yeniden derlenipilen çeşitli bilgiler, birçok insanı endişeye düşürmektedir. Bu konuda

Avrupa'dan çok sayıda protesto yükselmektedir.

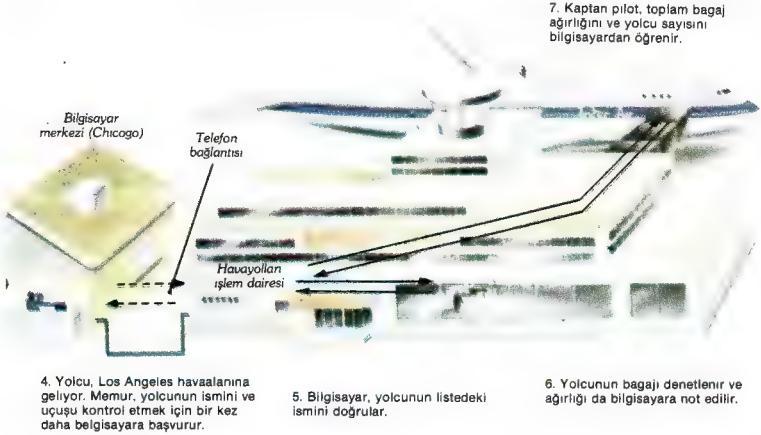
Kelime işlemci: Kelime işlemci, yazılı belgeler hazırlamak ve işleme sokmak için hazırlanmış bir bilgisayar sistemidir. Bu belgeler bir kez sisteme kaydedildikten sonra, ekrana çağırılıp yeniden düzeltilir. Kullanıcı bazı sözcükleri değiştirebilir, ekleyebilir, ya da çıkarabilir. Yazım yanlışlıkları düzeltilir, ya da bütün bir paragraf değiştirilebilir. Eğer belge yazılı olarak isteniyorsa, sistem bir yazıcıya bağlanabilir; ya da başka bir zaman kullanılmak üzere saklanabilir.

Kelime işlemciler, şimdiden yüzlerce ofiste kullanılmaya başlandı. Benzer şekillerde yazılacak olan yüzlerce mektubu yalnızca bir kez yazmak yeterli oluyor. İsim, adres gibi detaylar ise sonradan sistem üzerinde yazıya eklenebilir.

Kelime işlemciler, tekrar tekrar aynı şeylerin yazılması gereken kayıtların tutulmasında

Dünyayı Taramak

KÜÇÜKLÜ büyüklü birçok bilgisayar iş yerlerinde, fabrikalarda, okullarda, hastanelerde, inşaat şirketlerinde, rafineelerde ve daha sayılamayacak kadar çok yerde kullanılmaktadır. Bunlar verileri toplar, sıraya koyar, ya da işleme sokar. Bütün bu işleri insanlar için yapar. Basit işler için küçük kapasitede bilgisayarlar kullanılır. Ama daha karmaşık işler için daha büyük kapasiteli bilgisayarlar kullanılır. Bu bilgisayarın yüzlerce yan ünitesi olabilir. Bu üniteler bütün bir binaya, hatta bütün dünyaya



Hava alanlarındaki büroların yaklaşık tümü, yine alanda yer alan "ana bilgisayar"a bağlıdır. Bu ana bilgisayara, uçağa binecek yolcuların dökümü, bagajlarının kaç kilo geldiği hesaplanarak verilmiştir. (Üstte)

Yaklaşık her mikro bilgisayar, bir yazıcı ünitesine bağlanabilir. Yazıcı üniteleri de, ana sisteme bağlıdır. (Sağda)

büyük kolaylık sağlıyor. Ayrıca sürekli üzerinde değişiklik yapılması gereken belgeler de bu sistem yardımıyla fazla sıkıntıya girmeden yazılabiliyor. Örneğin bir avukatın, davalarıyla ilgili yazdığı yazılarda sürekli bir takım değişiklikler yapması gerekiyor. Bu gibi durumlarda bütün sayfaları yeniden teker teker yazmak yerine bilgisayar üzerinde yapılacak değişiklikler gösteriliyor. Gerisini bilgisayar tek başına yapıyor.

Havayollarında: Ulaşım araçlarında, bilgisayara çok gereksinim vardır. Bunlardan en çok havayolları bilgisayar kullanır.



Diğer ulaşım araçları henüz bilgisayar kullanabilecek kadar gelişmemiştir. Ancak yurdumuzda bazı demiryolu istasyonlarında

da bilgisayar kullanılmaya başlanmıştır. Bu da özellikle gişedeki memura büyük kolaylık sağlamıştır.

Yarının Dünyası

BİLGİSAYARLAR, aynı işte çalışan insanlardan çok daha verimlidirler. Bu yüzden çok yakın bir

zamanda bu tip işleri ya bilgisayarlar, ya da mikroişlemciler tarafından yönetilen otomatik makineler gerçekleştirecek.

Pek yakında...



Birçok satış yeri bilgisayarla sipariş alacak. Satışlar, otomatik olarak fişlere işlenecek.

TV'ye bağlı olarak çalışan bir video-terminal, gazete ve magazinlerin yerini alacak.

Arabalar, bilgisayar kontrolündeki makineler tarafından seri halinde üretilecek.

İnsan seslerini tanımak üzere hazırlanan programlar, otomatik kapı kilitleme sistemini de sağlayacak. Bu şekilde yalnızca istenen insanlar içeri alınacak.

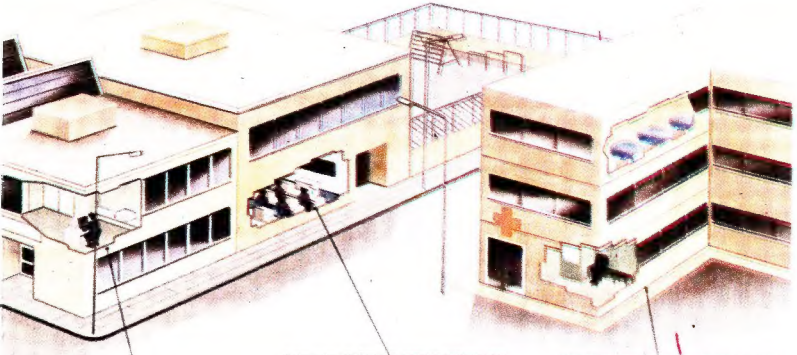
Günümüzde ve gelecekte bilgisayarların ve mikroişlemcilerin kullanılacağı bazı alanlar. (Üstte)

Yeni İşler: Bazı işlerin bilgisayarlar tarafından yapılması, birçok avantajı beraberinde getiriyor. İnsanlar için tehlikeli olan

yarak öğrendikleri meslekleri, bir anda bilgisayara bırakılmaktadır. Bu elemanlar da, açıkta kalmaktadırlar.

ların durumunu özellikle etkiliyor.

Bilgisayarlar da, birçok yeni mesleği beraberinde getiriyor.



Okullarda, öğrenciler bilgisayara alıştıkça, onun eğitimdeki rolü artacak ve daha çok eğitim programları hazırlanacak.

Bilgisayarlar, sağlık hizmetlerindeki alışıl gelen işleri aldıkça; bu hizmet dalı da gelişecek.

Ofislerde, çoğu kâğıt işleri bilgisayar terminalerinden yapılacak olan elektronik haberleşme sistemiyle gerçekleştirilecek. Bazı insanlar her gün işyerine gitmek yerine evlerinde çalışabilecekler.

Fabrikalarda, mal taşımacılığını insanlar yerine elektronik kontrollü robotlar yapacak.

bir takım işler, bilgisayar tarafından kontrol edilen makineler tarafından yapılacak. Ayrıca yine insanlar tarafından yapılmak istenmeyen basit, tekrar yapılacak işler de bilgisayarlar tarafından yürütülebilir.

Ancak, bu tehlikeli ve yinelenen işlerin yanında bilgisayarlar, bazı kalifiye işleri de üstlenebiliyorlar. Örneğin mühendislik işlerinde ve çok kaliteli seri üretimlerde bilgisayar kullanımı gün geçtikçe artmaktadır. İnsanları hem parasal yönden, hem de meslek açısından doyuran birçok iş de bilgisayar tarafından yapılmaktadır. Onların yıllarını harca-

Yeni Sorunlar: Basım endüstrisi, bilgisayarların bir iş kolunu nasıl etkilediğini gösteren güzel bir örnektir. Yazarın verdiği yazıları kâğıt üzerine basılabilecek hale getirmek artık özel yetişmiş eleman gerektirmiyor. Dizim ve basım işlerine bilgisayarın katılması, bu gibi işleri piyasada geçersiz kılıyor. Bu yüzden bu işleri yapan elemanlar, ellerindeki işin gitmesinden hiç de memnun değiller. Sendikaları aracılığıyla işlerinin ellerinden alınmamasına, yani bilgisayarın kullanılmamasına çalışıyorlar.

Günümüzde çok sayıda insanın işi ofislerde daktiloluk ya da bir takım yönetsel yazıları yazmaktır. Yazılı belgeleri saklayıp gerektiğinde basılı olarak çıkaran kelime işlemciler ise, bu insanların yerini almakla meşgul. Her ne kadar o bilgisayarları kullanacak olan kişiler de gereksinim olsa da, bu sayıda bilgisayarın işinden ettiği insanların sayısına kıyasla çok az. Bu durum, geleneksel bir şekilde sekreterlik yapan bayan-

Yeni tip bilgisayarlar üretmek, onları kullanmak ve yeni programlar yazmak için çok sayıda insan gerekiyor. Ofislerde, fabrikalarda, bu sistemleri işletecek elemanlara ihtiyaç var. Ancak, herkes mesleğini değiştirmek istemeyebilir. Bütün bir gün boyunca da bir ekranın başında çalışmak eğlenceli gelmeyebilir.

Boş Zaman mı, İzzetlik mi? Kuşkusuz bilgisayarlar insan yaşamını oldukça kolaylaştıracaklardır. Bilgisayarla, daha iyi bir sağlık hizmeti, daha iyi eğitim ve daha güvenli bir iş alanı demektir.

Şimdi çok sayıda insana gereksinim duyulan bazı işler, çok daha az sayıda insanın yardımıyla bilgisayarlar tarafından gerçekleştirilecek. Bunun sonucu ne olacak?

Birçok insan işsiz kalırken, çok az sayıda insanın doyurucu bir işi olabilecektir. Başka bir seçenek olarak da, bu işleri paylaşılacak. Böylece herkesin kendine ayıracak bol bol zamanı olacak.

İnsanlar ve Tarihler

AŞAĞIDAKİ kronolojik olaylar listesinde, bilgisayarı bugünkü gücüne getiren önemli gelişmelerden birkaçı sergileniyor.

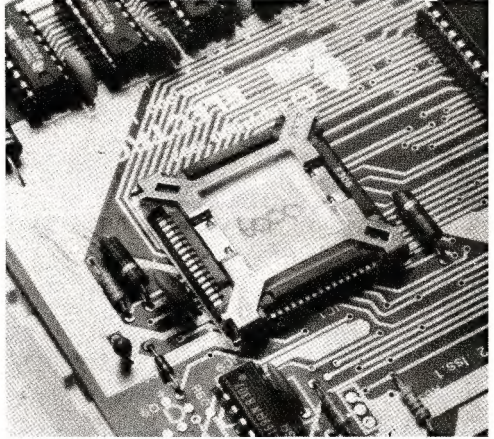
1617-Logaritmanın bulucusu John Napier 'kemiklerinin' buluşunu açıkladı. Bu kemikler, çarpma ve bölme işlemleri için hazırlanmış numaralı çubuklardan başka bir şey değildi. Bu 'kemikler', işlem yaparken kullanılan ilk mekanik aletlerden biriydi.

1642-Dünyanın ilk hesap makinesi Blaise Pascal adlı bir Fransız tarafından gerçekleştirildi. Bu makine, çarklar ve zincirlerle birbirine bağlanmış ve numaralanmış bir dizi tekerlekten oluşuyordu. Sayılar, makineye bu tekerlekler aracılığıyla veriliyordu. Ayrıca makinenin gerektiğinde sayıyı bir soldaki tekerleğe aktarma yeteğiği de vardı.

1694-Gottfried Leibnitz adlı bir Fransız matematikçi, bu hesap makinesini geliştirdi. Bu makine toplamadan başka çarpma, bölme, hatta karakök hesabı bile yapabiliyordu. Bu işlemleri, ardarda toplama metoduyla yapıyordu ki modern bilgisayarlar da aynı sistemi kullanır.

1801-Bir Fransız mühendis, Joseph Jacquard, sonsuz uzunluktaki bir delikli kart kemeri tarafından kontrol edilen bir ipek tezgâhı gerçekleştirdi. Kartlar, basınçlı hava yardımıyla "okunuyordu". Kontrol yardımıyla, renkli ipliklerin dokunması gerçekleştiriliyordu.

1835-Charles Babbage, 'analitik motor' üzerindeki çalışmalarını tamamladı. Bu makine, karmaşık problemleri çözmekle kalmıyor, aynı zamanda programlanabiliyordu da. Bu şekilde daha önce elde edilen sonuçlara göre hangi yola gideğini kendisi



saptayabiliyordu. Veriler ve programlar, makineye delikli kartlarla veriliyordu.

1946-Pennsylvania Üniversitesinde ENIAC tamamlandı. Bu makine, 1944 yılında gerçekleştirilen en son makineden 1000 kat daha hızlıydı ve 150 kW'lık güç harcıyordu.

1947-EDVAC tamamlandı. EDVAC, veri ve program saklanması amacıyla içinde elektronik bir bellek taşıyan ilk bilgisayardı.

1951-UNIVAC 1 yapıldı. Bu kez ilk defa bir bilgisayar yalnızca uzun ve karmaşık problemleri yapmakta kullanılmıyor, aynı zamanda veri ve program saklamak için de kullanılıyordu.

1955-Elektronik vanaların yerine kullanılmak üzere, belleklerde manyetik birim üniteleri gerçekleştirildi. Bu birimler, bir öncekilerden çok daha hızlı çalışıyor ve aynı zamanda çok daha az güç harcıyor. Çabukluğu nedeniyle bu birimler, bilgisayarı işlem hızını oldukça arttırmışlardır.

1957-Bilgisayarların ikinci aşamalı ortaya çıktı. Bunlar,

vanaların yerine transistörler kullanılıyorlardı. Transistörler, diğerlerine göre daha sıkı oldukları için, küçük bir yere daha fazla işlem gücü yüklenmesini sağlıyorlardı. Ayrıca çok az elektrik harcadıklarından, yeni bilgisayarların soğuması da eskilerine oranla daha kolay oluyordu.

1967-Üçüncü aşamalı bilgisayarlar piyasaya çıktı. Bunlarda, elektronik parçalar ayrı ayrı olarak değil de, bir bütün halinde entegre devre olarak kullanılıyordu. Bu bilgisayarlar, ikinci aşamalıdan daha da sıkı ve hızlıydılar.

1971-İlk mikroişlemci, dünyaya tanıtıldı. Bu mikroişlemci, bütün işlem elemanlarını içine alan tek bir entegre devreden oluşuyordu. Mikroişlemciler, ucuz mikrobilgisayarların yolunu açtılar. Onların ardından da 1980'lerin süper-hızlı makineleri geldi. Direkt giriş sağlayan klavyeler ve diğer giriş elemanları, bilgisayar kullanıcılarının veri ve komutları normal rakamlar ve sayılarla girmesine olanak verdi. Bilgisayarın içindeki yazılım sistemi, bütün bu karakterleri otomatikman alıp ikilik düzene çeviriyor. İkilik düzene çevrilmiş komutlar, işleme girmeye hazır oluyor.

Yaşanan Olaylar

XVIII. yüzyılda Charles Babbage adlı bir İngiliz matematikçi bir bilgisayar üzerine çalıştı. Ancak zamanın teknolojişinin yetersizliğinden dolayı düşüncelerini uygulama alanına geçiremedi.

Bilgisayarlar, sahne ışıklarını söndürmek için de kullanılıyorlar. Bir mikroişlemci ışıkları yakıp söndürmek, azaltmak ve gerektiğinde rengini değiştirmek üzere programlanıyor.

Amerika Birleşik Devletleri'nin 1880 yılı sayımı o kadar uzun sürdü ki, hükümet on yıl sonra yapılacak olan sayımda bilgisayar kullanmaya karar verdi. Seçtikleri makine delikli kart kullanıyordu. Bunlara kart sağlayan şirket, sonradan dünyanın en büyük işletmelerinden biri oldu (IBM).

İlk mekanik hesap makinesi, on yedinci yüzyılda Fransa'da satışa çıkarıldı. Şimdi orada hesap makinesi anlamına gelen 'calculator' kelimesi, o zaman çarpım tablosunu hazırlamak gibi matematiksel işlemler üze-

rinde çalışan insanlar için kullanılırdı.

İngilterede, polis in iki buçuk milyonluk parmak izi kayıtlarını bilgisayarda saklanabilecek bir forma çevirmek için uzmanlar beş yıl harcadılar. -

Bir mikroişlemcide bütün elektronik olaylar çipin yüzeyinde yer alır. Aktif olan kısmın kalınlığı, bir sabun köpüğünün kalınlığından daha azdır.

1979 yılında Fransa'da yapılan Tour de France bisiklet yarışında, puanları yarışı peşisıra izleyen bir kamu oyunun üzerine yerleştirilmiş bir bilgisayar hesaplıyordu.

Hesap cetveli, 1621 yılında İngiltere'de yapıldı. On dokuzuncu yüzyılın sonuna kadar İngiltere dışında pek az kullanıldı.

İnka'lar, sayısal kayıtlarını iplerin üzerine düğüm atarak tutarlardı. Çünkü yazılı dilleri yoktu. İnka'lar, onluk düzeni kullanmışlardır.

Rusya'da kullanılan abaküs, Rus piyasasına göre hazırlanmıştı. Yani çubukların üzerindeki

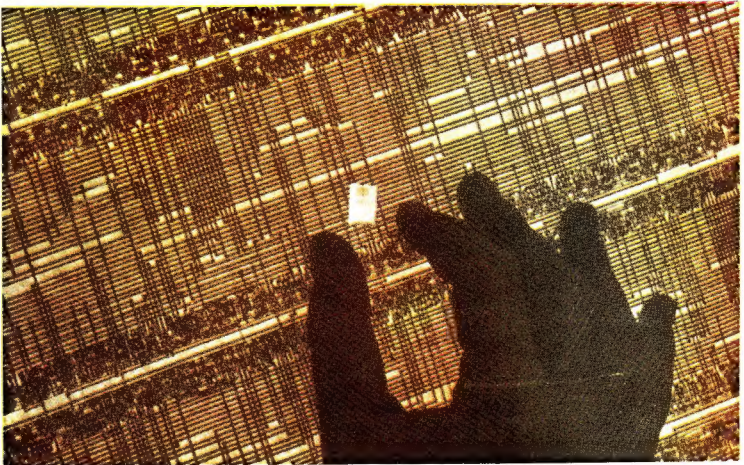
boncukların sayısı piyasaya uygun bir şekilde düzenlenmişti.

İkinci Dünya Savaşı sırasında İngiliz gizli servisi düşman şifrelerini çözmek için bir bilgisayar kullanmıştı. COLOSSUS adındaki bu bilgisayarın varlığı otuz yıl kadar öğrenilemedi.

Her yüz silikon çipten beşi, kullanılabilecek kalitede çıkmaktadır. Geri kalanı hemen atılacak kadar kalitesizdir.

Dünyadaki ilk me işlemleri, Amerika Birleşik Devletleri Kuvvetlerinde gerçekleştirilmişti. Amacı, ölen askerlerin ailelerine seri bir şekilde mektuplarını göndermekti.

Daktilo makinesindeki tuşlar, yazıcıları yavaşlatmak amacıyla hazırlanmıştır! Daktilo ilk kullanılmaya başlandığında, yazıcılar daktilo mekanizmasının çalışmasından daha çabuk yazıyorlardı. Bu yüzden bütün tuşlar birbirine giriyordu. Bu problem, yeni elektronik daktilo makinelerinde ortadan kalkmıştır. Şimdi, klavyedeki tuşlar yeniden düzenlenmiştir.



İÇİNDEKİLER

Sayılar ve Sayı Düzenleri	4
Hesap Makineleri	6
Bilgisayara Giriş	8
Boolean Cebiri	12
VEYA Kapısı	13
VE Kapısı	13
Değil Kapısı	13
Kapıların Birlikte Kullanılması	13
Çip	14
Çipin Yapımı	14
Bileğinizdeki Çip	16
Doğruluk Tabloları	17
Boolean Cebirinin Kuralları	17
Sayı Düzenlerini Çevirme	20
Bilgisayar ve Aritmetik	22
Flip-flop	28
Bilgisayar Donanımı	30
Bahis Nedir	35
Bilgisayarda Her Şey Adım Adım	38
Bilgisayar Kullanımı ve Programlanması	40
Bilgisayarlarla İletişim Kurma	42
Bilgi Çıkışı	46
Görsel Bilgi	48
Benzer ve Farklı	50



Akış Diyagramı	53
Programlamaya Giriş	57
Programlama Dilleri	63
Anında Çeviri	64

BASIC Diline Giriş	68
Donanım ve Kullanım Bir Arada	73
Şu Kuralı Anımsa	85
Bilgisayar Kahramanları ve Canavarları	90
Yıldız Kalesinin Doğuşu	91
Bilgisayar Oyunlarını Kendi Yararımıza Oynayalım	95
Akıllı Makinelerle Dolu Bir Dünya	98
Hesap Makineleri	100
Mikroişlemciler	102
Mikrobilgisayarlar	104
Konuş Benimle	107
Bilgisayara Görmeyi Öğretmek	114
Bilgisayar Gözüyle Dünya	116
İlk Elektronik Resim	116
Alfabe Ressamları	117
Elektronik Kitap	117
Bilgisayar Hollywood'a Giriyor	118
Dördüncü Boyuta Doğru Bir Pencere	120
Kaybolan Boyut	120
Müzik Makinesi	120
Dünyayı Taramak	122
Yarının Dünyası	124
İnsanlar ve Tarihler	125
Yaşanan Olaylar	126

